OpenTopography: A Services Oriented Architecture for Community Access to LIDAR Topography

Sriram Krishnan, Viswanath Nandigam, Christopher Crosby, Minh Phan, Charles Cowart, Chaitan Baru
San Diego Supercomputer Center, UC San Diego, 9500 Gilman Dr MC 0505
La Jolla, CA 92093, USA
{sriram, viswanat, ccrosby, mnphan, charliec, baru}@sdsc.edu

Ramon Arrowsmith
School of Earth and Space Exploration
Arizona State University
Tempe, AZ 85287, USA
ramon.arrowsmith@asu.edu

Abstract — High-resolution topography data acquired with LIDAR (Light Detection and Ranging) remote sensing technology have emerged as a fundamental tool for Earth science research. Because these acquisitions are often undertaken with federal and state funds at significant cost, it is important to maximize the impact of the data by providing online access to a range of potential users. The National Science Foundation-funded OpenTopography Facility hosted at the San Diego Supercomputer Center, has developed a cyberinfrastructure-based solution to enable online access to Earth science-oriented high-resolution LIDAR topography data, online processing tools, and derivative products. Leveraging high performance computational and data storage resources available at SDSC, OpenTopography provides access to terabytes of point cloud data, standard digital elevation models, and Google Earth image data, all co-located with computational resources for higher-level data processing. This paper describes the motivation, goals, and the technical details of the Services Oriented Architecture (SOA) and underlying cyberinfrastructure platform implemented by OpenTopography. The use of an SOA, and the co-location of processing and data resources are unique to the field of LIDAR topography data processing, and lays a foundation for providing an open system for hosting and providing access to data and computational tools for these important scientific data, and is an example for similar data and knowledge community-oriented cyberinfrastructure systems in other domains.

Keywords: LIDAR, SOA, Cyberinfrastructure

I. INTRODUCTION

High-resolution topography data acquired with LIDAR (Light Detection and Ranging) remote sensing technology have emerged as a fundamental tool for Earth science research, as well as infrastructure planning and engineering applications. These data, typically collected from an airborne platform, represent the Earth’s surface, overlying vegetation, and urbanized environment in 3 dimensions, with unprecedented detail over large spatial extents (see Figure 1). As a result, these datasets allow scientists to study the processes that act at the Earth’s surface at resolutions not previously possible, yet essential for their appropriate representation. Over the past ten years, LIDAR data have provided new insights into geologic phenomena ranging from earthquake hazards to ice sheet dynamics (e.g. [2], [3], [4]) with new applications and analysis techniques continuing to emerge.

Figure 1. LIDAR point cloud data from the Old Faithful area of the Yellowstone National Park, WY [1]. Individual LIDAR returns are colored by relative elevation (blue – low, red – high) and reflectivity of the material they were returned from. Buildings, vegetation, vehicles, as well as the Old Faithful geyser are evident from the figure.

LIDAR is a remote sensing technology that combines a high-pulse rate scanning laser with a differential global positioning system (GPS), and a high-precision inertial measurement instrument on an aircraft or helicopter to record dense measurements of the position of the ground, overlying vegetation, and built features [5]. Firing up to several hundred thousand pulses per second, LIDAR instruments can acquire multiple measurements of the Earth’s surface per square meter over hundreds to thousands of square kilometers. The resulting dataset, a collection of measurements in geo-referenced X, Y, Z coordinate space known as a “point cloud”, provides a 3-dimensional representation of natural and anthropogenic features at fine resolution over large spatial extents.

High-resolution topographic data collection for research, planning, and regulatory activities is burgeoning. Because these
acquisitions are often undertaken with public funding by federal, state, and local agencies at a cost of several hundred dollars per square kilometer, it is important to maximize the impact of the data by providing easy online access to a range of potential users. However, the volume of data generated by LIDAR technology often impedes their wider dissemination and utilization. Many organizations that acquire LIDAR topography struggle to manage the data due to their large size, and the lack of storage capacity, bandwidth, and expertise necessary to make these data available for online community access in an efficient manner.

Figure 2. Idealized LIDAR collection and processing workflow, from data acquisition to processing and delivery. The OpenTopography Facility provides online access to raw point cloud datasets and derivative data products, such as full-featured and bare earth Digital Elevation Models (DEM).

To address the challenge of open online access to these valuable scientific data and corresponding processing tools, the OpenTopography Facility (http://www.opentopography.org) hosted at the San Diego Supercomputer Center (SDSC), has developed a cyberinfrastructure-based solution to enable online access to Earth science-oriented high-resolution LIDAR topography data, online processing tools, and derivative products (see Figure 2). OpenTopography provides a production example of an end-to-end multi-tiered cyberinfrastructure for scientific data and knowledge management for a diverse community.

The underlying cyberinfrastructure platform that powers OpenTopography was originally developed as part of the National Science Foundation (NSF)-funded GEON Project [7] as an R&D application called the GEON LIDAR Workflow (GLW) [8, 9]. The success of the GLW illustrated the need for a production-quality online portal to serve the rapidly growing community of Earth science LIDAR users. OpenTopography has since been funded as an NSF data facility with an active community of several thousand scientists, educators, students, and government agency users. OpenTopography’s data holdings are currently focused on active earthquake faults in the western United States, and the system has been used to access and process data that have been used in many publications related to active tectonics and earthquake hazards [10, 11, 12].

It should be noted that Web portals that enable access to LIDAR data are not without precedent. For instance, the NOAA Digital Coast Data Access Viewer [13] provides a Web-based interface to data managed by the NOAA Coastal Services Center. However, there are no existing systems that provide co-location of LIDAR data and processing to the extent envisioned by the OpenTopography Facility. Furthermore, OpenTopography is alone in its vision of providing access to powerful back-end storage and compute resources via open Application Programming Interfaces (APIs).

The technical goal of the OpenTopography Facility is to build a modular system that is flexible enough to support the growing needs of the Earth science LIDAR community. In particular, we strive to host and provide access to datasets as soon as they become available, and also to expose greater application level functionalities to our end-users, e.g. generation of custom Digital Elevation Models (DEMs), and hydrological modeling algorithms for generation of derivative products. Furthermore, we envision enabling direct authenticated access to our back-end functionality through simple Web service APIs, so that users may access our data and compute resources via clients other than Web browsers.

To address the above technical goals, OpenTopography has developed a scalable Service Oriented Architecture (SOA) that provides a strong foundation upon which future facility developments will be based. This paper discusses the underlying OpenTopography cyberinfrastructure that supports Web based LIDAR point cloud data access and a range of downstream processing and analysis tools for users.

The rest of the paper is organized as follows. In Section II, we provide an overview of the OpenTopography architecture and functionalities, followed by a detailed discussion of the implementation details for the various components within the system in Section III. In Section IV, we discuss the process by which new data and processing capabilities are added to the OpenTopography system. We present ongoing and future work in Section V, and finally, we conclude with a discussion of the impacts of the OpenTopography facility in Section VI.

II. OPENTOPOGRAPHY OVERVIEW

In order to facilitate access to LIDAR data for users with varying ranges of expertise and computing resources, OpenTopography provides data access for different levels of user sophistication (see Figure 3). At one end of the spectrum, expert users can access entire point cloud datasets, often many billions of individual points, and run a particular processing routine with customized parameter settings. Typically this processing involves the generation of a customized DEM using the processing tools provided by OpenTopography. The system also permits users to produce derivative products such as common geomorphic metrics like slope, as well as visualizations of the products. In the middle of the spectrum, pre-computed DEMs at resolutions that are optimal to the datasets and organized into manageable tiles are a popular data product. At the other end of the spectrum, non-expert users can access pre-processed network linked Google Earth images that provide streaming access to close to terabytes of LIDAR-derived imagery [6] for synoptic data browsing.
Service Oriented Architectures (SOAs) have several features that are beneficial for building cyberinfrastructure solutions for science and engineering [14]. Through encapsulation of back-end functionality, SOAs reduce the complexity of the processes that are part of the scientific workflow. Applications and data resources can be wrapped as Web services, exposing only a thin API to clients. The implementation details and the infrastructure back-end can be completely hidden from end-users scaled up without any change to the service APIs. Web service APIs also enable interoperability across systems and architectures through the use of open standards, such as XML and SOAP [25]. Clients may be written in a variety of languages, and run on disparate platforms. Due to these benefits, and the goal of making OpenTopography an open and scalable community resource, we have implemented the system as an SOA.

The details of the multi-tier SOA implemented by the OpenTopography (Figure 4) are as follows:

**Infrastructure Tier:** This tier consists of the server resources that host the data, processing tools, and services. To efficiently manage and store the LIDAR point cloud data delivered in ASCII format, we use a customized multi-node partitioned IBM DB2 database. LIDAR datasets in the ASPRS LAS format [16] are hosted on a separate dedicated server with large disk capacity. In addition, we have a large SMP machine with more than 192GB of memory to run computational codes for higher-level processing, and a dedicated server to run visualization software such as Global Mapper [17].

**Services Tier:** Access to the infrastructure tier is provided to users via the Services tier. Every software component is wrapped as a Web service, accessible via a SOAP API. Available Web services include data selection from our ASCII DB2 database and LAS systems, generation of digital elevation models, and creation of derivate products and visualizations. A security layer enables authenticated access to these services.

**Applications Tier:** Most users access the functionality provided by the Services Tier via the OpenTopography portal. The portal enables a user to run pre-defined workflows that harness the services provided by the system. An easy to use graphical user interface guides users through the data selection and processing parameterization steps. Other clients include the Opal dashboard, described in Section III, and command-line programs. In the future, we expect that services will be accessed via scientific workflow tools like Kepler [18], and GIS software plug-ins (e.g. ArcGIS [19]).
III. IMPLEMENTATION DETAILS

A. Infrastructure Back-end

The infrastructure tier encompasses the data management systems for hosting LIDAR point cloud datasets and compute resources for processing and visualization of these datasets.

Partitioned Database System: ASCII formatted LIDAR point cloud datasets are stored in an 8-node partitioned IBM DB2 database. The data are stored across these multiple partitions that function together as a single database system. [32]. A partitioned database allows us to take advantage of the cumulative processing power and memory distributed across multiple commodity clusters to efficiently manage these massive datasets at a reasonable cost [31].

Each individual dataset is distributed across multiple database tables with its own schema and structure to accommodate the point returns and attributes specific to the dataset. The number of tables varies depending on the dataset’s spatial extent and density distribution.

![Figure 5. Distribution of datasets of varying sizes across database partition groups. Smaller datasets are spread across fewer database partitions.](image)

We create multiple partition groups, which are a set of one or more database partitions, of varying sizes to accommodate the different sized datasets, as shown in Figure 5. Small datasets like the Eastern California Shear Zone (less than 20GB in size) are assigned to smaller partition groups, whereas the larger datasets such as the GeoEarthScope Northern California LIDAR project (close to 1TB) are assigned to the larger partition groups. The assignment of datasets to partition groups is based on heuristics, and gives us optimal performance, with respect to the overhead. Data are distributed across partitions using hash partitioning. Uniform data distribution across all partitions provides the best parallelization and performance for selects, which form the majority of the queries we run, and can be achieved by choosing the appropriate partitioning key. The attribute with the most distinct values serves as the best choice for the partition key. Some LIDAR datasets contain a GPS timestamp attribute, which is ideal for use as a partitioning key. For datasets that lack attributes or do not have distinct attributes, depending on its spatial coverage, we can choose either the latitude or longitude of the point return as a partition key.

We use regular (B-tree) indexes on the latitude and longitude values of the point cloud return. In a partitioned environment, a query can perform efficiently using these indices. We also incorporate a metadata table to further improve queries. The metadata table stores information about each of the tables including data extent, number of rows in the table, and coordinate system. A user query in the form of a latitude-longitude bounding box is first applied to the metadata table to obtain a subset of tables whose extents intersect with the bounding box. The spatial sub selection query then runs only on that subset of tables to return the requested data in ASCII format. We also perform additional database level optimizations to get the best possible performance for serving these point cloud datasets. Details of the database organization and optimization techniques are beyond the scope of this paper, and can be found elsewhere [30].

Currently, our LIDAR point cloud database is approximately 3.5 TB in size and contains nine datasets with more than 37 billion LIDAR point cloud returns.

LAS System: The ASPRS LAS binary format is the industry standard for LIDAR data exchange. LAS files store data more efficiently than ASCII files, and also include a header that contains metadata information about the file including coordinate system, extent, and number of points.

![Figure 6. The hybrid database and file-based LAS management architecture. The point cloud data files are stored on the file server in LAS format, but the metadata for the LAS files is stored in a relational database for faster querying.](image)

For the management of LAS data, we use a hybrid approach with an IBM DB2 database that serves as a metadata repository, with the LAS files stored on a file server. The database contains metadata tables, one for each LAS dataset in the system. The table is populated using information from the LAS file headers when the data arrive at OpenTopography, and includes the name and location of the file in the system, the data extents, and number of points.

When a user’s bounding box query is submitted, it is applied against the metadata table, and the LAS files whose extents intersect with the bounding box are returned. The
system then extracts the required data from these files using open source software for querying LAS data ([34], [36]).

**Compute and other Resources:** In addition to providing online access to LIDAR point cloud data, OpenTopography also provides Web-based on-demand processing services that allow users to generate useful derived products from the data. These services include DEM generation using a LIDAR-specific gridding algorithm, DEM format translations, generation of hill shade and slope grids, and product visualizations. We provide this processing by leveraging high performance hardware available at SDSC. The DEM generation process is significantly faster if the whole grid can be stored in memory, and thus a large memory compute resource is important. We utilize a Sun X46 machine with eight core processors and 192 GB of memory for these computations. The resulting derived products are stored on a high performance SAN disk connected to the machine via a high-speed fiber channel interface. We also have a dedicated Windows server hosting Global Mapper [17], which is a Windows based GIS software package for the generation of hill shades and Google Earth KMZ files for data browsing via a Web browser.

To track jobs run by the users, we also have a separate job-monitoring database hosted on a shared resource. The job-monitoring database contains information about the user’s authorization level, details of past jobs, along with performance metrics such as duration of each processing step. A Web interface called myLIDAR Jobs uses the monitoring database to allow users to track and resubmit previous jobs. The database also allows OpenTopography staff to identify system problems and improve performance by examining the job timings and failures, as well as report on usage.

**B. Services Middleware**

The services tier enables access to the computation and data resources on the infrastructure tier via simple SOAP APIs. In order to build the Web services, we have leveraged the open-source Opal toolkit ([24], [26]) developed by the National Biomedical Computation Resource (NBCR) [23] and SDSC.

Opal is a toolkit that enables users to automatically wrap scientific applications running on cluster and Grid resources as Web services, and provides Web-based and programmatic access to them, without writing a single line of Web service code. Developers simply write an XML-based configuration file for an application, and deploy it as a Web service using Opal’s command-line deployment tool.

The Opal toolkit has several features that are advantageous for OpenTopography. Installation and subsequent service deployment are straightforward and can be accomplished in a matter of hours for each new application. Opal provides mechanisms to manage and track job submissions, with the help of a back-end database. It allows monitoring of job and system status by providing charting tools out of the box. It also provides a “Dashboard”, which can be used by administrators to track system and usage information (see Figure 7), and access automatically generated user interfaces that can be used for prototyping and debugging. Finally, Opal provides a number of ways to ensure that only authorized users are able to access our Web services. In our current system, we have configured Opal to restrict jobs based on IP addresses. Because of the use of Opal, our software development time was significantly reduced compared to building Web services from scratch.

![Figure 7. Access to Job Statistics provided by the Opal Dashboard – usage statistics for various OpenTopography services over the past month can be seen above. Usage charts can be modified by (un-)selecting specific services and updating time intervals.](image)

OpenTopography applications exposed as Web services via the Opal toolkit are as follows:

**Point Cloud Access:** Access to point cloud data in ASCII or LAS format given a bounding box, dataset ID and other filtering parameters. Depending on user preference, the output point cloud could be in ASCII or LAS formats.

**Points2Grid:** Generation of DEMs using a local gridding method [33]. The service accepts input parameters such as grid resolution, search radius, input and output formats, and generates a custom DEM.

**Format Translation:** Translation of DEMs from Arc ASCII to GeoTIFF and IMG formats using the GDAL toolkit [27]. GeoTIFF and IMG formats are popular among users because the generated grids can be easily loaded into various GIS analysis software tools.

**Derivative Products:** Generation of slope and hill shade grids in GeoTIFF and IMG formats, to be downloaded into GIS analysis tools.

**Visualization:** Generation of hill shades, browser images and Google Earth image overlay files for visualization on a Web browser.

**C. The OpenTopography Portal**

OpenTopography users interact with the system via a Web portal built using the GridSphere framework [29]. An interactive Google Map interface showing the extent of available data allows the user to define a rectangular area of interest by drawing a box on the map. The coordinates of this
bounding box plus optional attribute type (e.g. ground points only) form the user’s query (DB2 Selection service).

Once a query has been defined, the user chooses their processing options. Point cloud data download for the user’s area of interest is available in ASCII or LAS binary formats. The user can also pass the subset of their selected point cloud to a series of services that will process the data on the fly to create custom DEMs and derivatives tailored to their science application.

![Figure 8. Typical OpenTopography workflow and related Web services. Starting with the point cloud selection, users may choose to customize runtime parameters and execute a set of available processing services.](image)

Using a high-performance local gridding algorithm (Points2Grid service), OpenTopography will produce a DEM to the user’s grid resolution and algorithm specifications. The user can also choose the format of the grid product that is generated (Format Translation service) so that the file is compatible with their preferred analysis software. Finally, the user is given the option to generate common geomorphic metric products such as a slope grid derived from the DEM (Derivative Products service).

Once the user has defined their processing requests, they give their job a name and description, and submit it. Most jobs take a few minutes to complete, and the user can either monitor their job using the personalized “myLIDAR Jobs” workbench (see Figure 9), or await an automated email notification that their job has completed. Once loaded, the metadata table gets populated accordingly with the table name, extents and number of points in the column.

![Figure 9. The “myLIDAR Jobs” workbench on the OpenTopography portal – users can monitor job status, view past results, and resubmit jobs with modifications to parameters or areas of interest.](image)

### IV. ADDING NEW DATA & APPLICATIONS

#### A. OpenTopography Data Ingestion

OpenTopography can ingest point cloud data in both ASCII and ASPRS LAS binary formats. Both the ASCII data and LAS files need to go through an ETL (Extract, Transform, Load) process before they can be made available for access.

**ASCII LIDAR data**: ASCII LIDAR data sent to OpenTopography are collected by a variety of vendors and delivered in non-standard formats. The initial ETL phase involves extraction, error checking, and uniform coordinate conversion, if necessary. This is followed by an organization phase, where the data is organized based on spatial extent and density. These files are eventually converted to tables in the database. Once loaded, the metadata table gets populated accordingly with the table name, extents and number of points in the column.

**LAS Binary data**: One of the advantages of data in a standard specification is the elimination of the data cleaning and validation stage. LAS data arrive in a single format with standard attributes. The LAS ingestion process invokes a pre-processing script that utilizes the libLAS libraries [34] to read the file headers and populate the metadata table in the LAS database with header information extracted from each of the files along with the name and location of the files on the system.

Irrespective of whether the data is in ASCII of LAS formats, access to them is provided via our standard Web service APIs.

#### B. OpenTopography Processing Tools

Every new processing tool that needs to be added to the OpenTopography system is installed on one of our resources on
the Infrastructure tier, and then wrapped as a Web service using the Opal toolkit.

After successful compilation and installation of the application, an Opal XML configuration file for that application is written. Using the XML configuration file, an application is then deployed as a Web service using Opal’s command-line deployment tools.

The schema for the Opal configuration file is shown in Figure 10. The only mandatory fields are the location of the application binary, which is used to run the application, and usage information, which displayed to an end-user. Additional textual description, application type (parallel or serial) and default arguments can be added. Finally, Opal also defines a schema that is used to describe the syntax of the command-line arguments [26], which is used by Opal to dynamically generate a Web interface on its Dashboard. We have found that we can define and debug the application configuration for a new application, in a matter of hours, reducing development and deployment time significantly.

New features currently being planned include a consolidated user interface for data discovery, and providing federated access to data hosted by organizations other than OpenTopography, such as the U.S. Geological Survey and NSF’s National Center for Airborne Laser Mapping. We plan to add additional algorithms for the generation of DEMs (e.g. streaming TIN [37]), and tools for hydrological processing (e.g. TauDEM [28]) to our available workflow. Also planned is enabling access to compute and data resources via workflow tools such as Kepler, GIS software such as ArcGIS, and other scripting interfaces via the Web service APIs. All of the above features can be implemented in an efficient manner due to the adoption of a SOA for OpenTopography.

In terms of processing, we are investigating the use of techniques such as Google’s MapReduce [35] for the parallel implementation of DEM generation, which would enable us to process larger grids at much higher resolutions [15].

We are also planning to investigate the use of external user defined functions (UDFs) within DB2 for querying LAS data. By running the UDFs in parallel across the partitioned shared-nothing database cluster, we can potentially improve the performance of the LAS queries.

VI. CONCLUSIONS

The NSF-funded OpenTopography Facility utilizes cyberinfrastructure technologies to enable online access to high-resolution LIDAR topography datasets, Web based processing tools, and derivative products. Making use of a Services Oriented Architecture (SOA) design, the OpenTopography Facility provides a production-quality infrastructure for data access at multiple levels - point cloud data and processing tools, pre-computed DEMs, and Google Earth streaming image files for visualization. Leveraging the high performance computational and data storage resources at SDSC, OpenTopography provides access to terabytes of data, all co-located with computational resources for higher-level data processing. All core components have been wrapped as Web services using the Opal toolkit. The GridSphere-based OpenTopography portal provides access to these Web services. Programmatic access to the Web services will be provided to our power users in the near future, so that the resources may be accessed via 3rd party tools. The use of an SOA, and the co-location of processing and data resources are unique to the field of LIDAR topography data processing, and lays a foundation for providing an open system for hosting and providing access to data and computational tools for LIDAR, and is an exemplar for similar data and knowledge community-oriented cyberinfrastructure systems.

V. ONGOING AND FUTURE WORK

Although the OpenTopography SOA provides a strong foundation for enabling access to LIDAR data products and processing capabilities, several improvements remain. Ongoing and future work for OpenTopography fall into two realms: 1) addition of new features and functionality for our end-users, and 2) improvements to our infrastructure with respect to both system performance and data growth.

We acknowledge previous and current GLW and OpenTopography team members, including Jeff Conner, Efrat Jaeger-Frank, Han Kim, Newton Alex, Santhanakrishnan Balakrishnan, and Javier Colunga. We also acknowledge the
OpenTopography Advisory Committee for their guidance and feedback.

REFERENCES

[1] EarthScope LIDAR data are provided by the Plate Boundary Observatory operated by UNAVCO for EarthScope (http://www.earthscope.org) and supported by the National Science Foundation (No. EAR-0350028 and EAR-0732947).


[34] libLAS, ASPRS LIDAR data translation toolset, http://liblas.org

