An Object Storage for Distributed Acoustic Sensing

Yiyu Ni*[1], Marine A. Denolle[1], Rob Fatland[2], Naomi Alterman[2], Bradley P. Lipovsky[1], and Friedrich Knuth[3]

Abstract

Large-scale processing and dissemination of distributed acoustic sensing (DAS) data are among the greatest computational challenges and opportunities of seismological research today. Current data formats and computing infrastructure are not well-adapted or user-friendly for large-scale processing. We propose an innovative, cloud-native solution for DAS seismology using the MinIO open-source object storage framework. We develop data schema for cloud-optimized data formats—Zarr and TileDB, which we deploy on a local object storage service compatible with the Amazon Web Services (AWS) storage system. We benchmark reading and writing performance for various data schema using canonical use cases in seismology. We test our framework on a local server and AWS. We find much-improved performance in compute time and memory throughout when using TileDB and Zarr compared to the conventional HDF5 data format. We demonstrate the platform with a computing heavy use case in seismology: ambient noise seismology of DAS data. We process one month of data, pairing all 2089 channels within 24 hr using AWS Batch autoscaling.

Cite this article as Ni, Y., M. A. Denolle, R. Fatland, N. Alterman, B. P. Lipovsky, and F. Knuth (2023). An Object Storage for Distributed Acoustic Sensing, *Seismol. Res. Lett.* **95**, 499–511, doi: 10.1785/0220230172.

Introduction

Distributed acoustic sensing (DAS) is an emerging technology for measuring seismic vibrations. DAS is revolutionizing geophysical sensing, because it provides unaliased wavefield images with resolution as fine as ∼0.5 m along optical fibers up to ∼170 km in length (Waagaard *et al.*, 2021). DAS utilizes repeated laser pulses along optical fibers to measure phase changes of light that are scattered from imperfections along the length of the cable. These phase changes are proportional to the rate of extensional strain along the axis of the fiber. Typical sampling rates using DAS are in the range of 0.01 Hz–100 kHz, with longer cable lengths ultimately limiting the highest possible sampling rate. DAS dramatically expands the capability of dense seismic observation and has been used for a vast range of applications, such as detecting new tectonic faults (Lindsey *et al.*, 2019), high-resolution subsurface imaging (Atterholt *et al.*, 2022; Yang, Zhan, *et al.*, 2022), cataloging earthquakes (Nayak *et al.*, 2021), tracking marine mammal vocalizations (Wilcock *et al.*, 2023), observing ocean gravity waves (Williams *et al.*, 2022), and monitoring vehicular traffic and infrastructure state of health (Lindsey *et al.*, 2020; Wang *et al.*, 2021). DAS has the potential to transform geophysical sensing in the broadest sense and push the frontiers of geophysical research.

DAS data rates vary widely depending on the observational target. Data rates may be calculated as $r = \alpha f L/\delta x$, in which α is the size of one sample on one channel, here assumed to be $\alpha = 4$ Bytes, f is the sampling frequency, L is the total fiber length, and δx is the spatial sampling interval (channel spacing). DAS observations targeted to measure ocean surface gravity waves (SGWs) need only be recorded at 1–10 s period sampling rate to capture unaliased T = 2–20 s period SGWs. Unaliased spatial sampling in the shallow water limit requires a spatial sampling $T\sqrt{gH}/2$, in which g is the gravitational acceleration, and H is the water depth. In 100 m deep water, 2–20 s period SGWs have wavelengths of ∼30–300 m; the same periods have wavelengths 100–1000 m in 1 km deep water. Assuming maximal DAS range ∼150 km and that samples are stored as single precision floating point numbers gives data rates from 50 MB/d to 1.7 GB/d. As an example of an intermediate data rate, we now consider DAS observations of tectonic earthquakes. Here, although seismic Q limits observable source frequencies to ∼10 Hz (Shearer, 2019), azimuthal variation in seismic observables (Kaneko and Shearer, 2015) and small-scale variation in near-surface structure (Spica *et al.*, 2020) may justify spatial sampling at 1–10 m; the total resulting

1. Department of Earth and Space Sciences, University of Washington, Seattle, Washington, U.S.A., https://orcid.org/0000-0001-5181-9700 (YN); https://orcid.org/0000-0002-1610-2250 (MAD); 2. eScience Institute, University of Washington, Seattle, Washington, U.S.A., https://orcid.org/0000-0003-4556-8321 (RF); https://orcid.org/0000-0003-0392-3890 (NA); 3. Department of Civil and Environmental Engineering, University of Washington, Seattle, Washington, U.S.A., https://orcid.org/0000-0003-1645-1984 (FK)

*Corresponding author: niyiyu@uw.edu

data rates are on the order of 14–28 GB/d, assuming a shorter, 10–20 km long urban optical fiber span. Other applications may result in much higher data rates. DAS observations targeted to observe the vocalizations of marine mammals may attempt to sample ~750 Hz signals with 2 m wavelengths (Douglass *et al.*, 2023) associated with the acoustic wave speed in water. Such observations result in data rates of 19.4 TB/d. Optical fading and other effects may, in practice, necessitate greater spatial sampling intervals (e.g., as in Wilcock *et al.*, 2023); for that reason, we take this number to be an upper bound estimate.

Managing 10–1000 TBs data volumes is presently the foremost barrier to democratizing DAS data access and research. This poses a challenge for storage and sharing on multiple fronts. First, the data volume is immense and surpasses today's total volume of seismic data stored at national archives (e.g., at Incorporated Research Institutions for Seismology Data Management Center [IRIS-DMC], Arrowsmith *et al.*, 2022). DAS metadata are largely unconventional and yet to be agreed upon by the community. Furthermore, DAS data sharing has been limited because, as of June 2023, no federally funded facility accepts DAS data, and because enabling public access to large datasets at an individual principal investigator (PI) level is difficult. A pioneering effort to address these needs led a collective of PIs to open a shared storage system for heterogeneous DAS datasets (Spica *et al.*, 2023). Similar to the Department of Energy's Geothermal Data Repository (GDR; Weers *et al.*, 2022), PubDAS uses GLOBUS endpoints to transfer entire DAS data files to end users. Beyond data storage, the processing of large volumes of DAS data is limited due to impractical choices in data formats (detailed subsequently), limiting the processing required for seismological research. New DAS data analysis frameworks have been developed, for example, DASCore that provides DAS data input/output (I/O) and basic processing (Chambers *et al.*, 2022), and DAS data storage and analysis that enables parallel data processing on modern supercomputers (Dong *et al.*, 2020).

Cloud computing has revolutionized scientific computing with large datasets. Seismological and, therefore, DAS research typically handles subsets of data using identical workflows and software. These workflows are perfectly suited for horizontal parallelization, the optimal design for cloud computing (MacCarthy *et al.*, 2020). The rise of cloud computing also fueled the development of data formats optimal for cloud-native storage and parallel throughput. The combination of cloud-native data formats and horizontal scaling of cloud computing architecture is an attractive solution for DAS seismological research.

This study proposes a novel data platform for DAS that incorporates object storage with cloud-optimized data formats: we store DAS data in Zarr (Miles *et al.*, 2020) or TileDB (Papadopoulos *et al.*, 2016) file formats using the MinIO open-source object storage. MinIO is compatible with Amazon Web Services (AWS) Simple Storage Service (S3) object storage. The combination of these technologies replicates on local Linux servers—a cloud-native framework that can be seamlessly ported to AWS. We demonstrate below with several canonical examples that typical DAS workflows can be designed with low memory, parallelized jobs on a local Linux server and AWS. The demonstration includes examples of public access to the new DAS public archive of the University of Washington (UW) FiberLab.

Background

A typical data flow in a DAS experiment begins at acquisition. DAS interrogator units usually have built-in storage capable of storing several TBs of data. Some installations may involve real-time data streaming, as was the case for the Whidbey and SeaDAS-N experiments at the FiberLab (Lipovsky, 2023a,b). Given the typical configurations with the number of channels and sampling rate, internal storage is insufficient for months-long continuous recordings that can accumulate up to hundreds of TBs. For this reason, it is common to use a separate network attached storage (NAS) connected to the DAS interrogator via ethernet to sync the data as the unit is recording. The NAS can also serve as the primary data repository for further processing, or data may be subsequently transferred to a separate archival server or workstation. This last step necessitates transferring full copies of the relevant HDF5 files, which could amount to hundreds of TBs and even petabytes (PBs) worth of data. This is the model, for example, of PubDAS and the GDR (Spica *et al.*, 2023). Still, the expectation is that data transfer will proceed with single-thread requests and be limited by the internet speed.

Cloud storage

Cloud storage is a general term referring to services that allow users to store data somewhere other than their local computers but with the maintenance and upkeep of such storage systems handled by the "cloud provider" (e.g., AWS) rather than the end user or IT personnel affiliated with the end user. In this arrangement, the cloud provider is responsible for storing, managing, and maintaining the infrastructure and network, ensuring that data are safely accessible with virtually unlimited capacity. However, all of this is provided at a cost, and, consequently, cloud providers will offer many storage models with variations in flexibility to stretch the end-user's dollar further. One such type is "object storage," which treats each data file as an object identified by globally unique IDs. Because there are no hierarchical structures across objects, this type of storage appears simpler than a traditional networked file system. It can be scaled up much larger for an order-of-magnitude reduction in cost. It is provided by most cloud platforms, usually under individual brand names (e.g., AWS "S3," Microsoft Azure's "Blob Storage," and Google Cloud Platform's "Object Storage"). Several do-it-yourself server products allow users to deploy local object storage systems. One example of these local

storage systems is MinIO. This high-performance open-source object storage implementation provides an "object storage" abstraction across various modes of deployment. The single-node single-drive is designed for a single machine, single drive, and datasets typically up to 20 TBs on a workstation. The single-node multiple-drive is designed for a single machine with multiple drives and datasets typically up to 500 TB on a rack-mounted server. Finally, the distributed multiple-node multiple-drive is designed for a multiple-node server and multiple drives with datasets typically at and above 1 PB on multiple units of rack-mounted servers. MinIO is AWS S3-compatible in that it uses an identical application programming interface, which supports access control through the credential keys and secrets granted to users with proper permissions. This compatibility facilitates researchers moving between commercial cloud storage systems and locally hosted options with minimal impact on their code or workflow.

Challenges in storing and formatting large seismic data

Traditionally, seismic data repositories have mostly been hosted by data centers managed partly by research institutions. One example is that the IRIS-DMC and its data archive in Seattle, Washington. The archive has grown over time and accumulated ~877 TB data as of 1 January 2023. The IRIS-DMC has shipped 7.2 PB data cumulatively between 1 January 1990 and 1 January 2023, and more than 1 PB data in the single year of 2022. Other seismic archives face similar challenges, especially driven by user-specific research and trends toward tackling larger and larger datasets (Quinteros *et al.*, 2021; Arrowsmith *et al.*, 2022). Supporting this level of data growth has already greatly challenged the DMC and has prevented the archive from hosting more voluminous DAS datasets.

Another example of a large seismic archive on the cloud is the AWS Open Data of the Southern California Earthquake Data Center (SCEDC; see Data and Resources), which has provided its entire archive of continuous time series and curated dataset, including an earthquake sequence dataset recorded on DAS stored in DAS-native format (Hauksson *et al.*, 2020; Yu *et al.*, 2021). The SCEDC has not changed the formats the seismic instruments provide, which suit cloud storage with small (~10 MBs) single-day, single-channel files. Another example of a large seismic archive is that of PoroTomo (Feigl *et al.*, 2016), which was collected in 2016, and is hosted in DAS-native SEG-Y and HDF5 formats as an AWS Open Data (see Data and Resources). The DMC is transitioning to cloud storage on AWS in 2023 and will adopt a new cloud-optimized data format for future incoming data. Thus, there has been a trend of moving data to the cloud where storage is available, albeit oftentimes at significant financial cost.

In seismology, variants of HDF5 schema (e.g., adaptable seismic data format [ASDF], Krischer *et al.*, 2016; HDF5eis, White *et al.*, 2023) have been introduced to accommodate multidimensional time series with no dataset size restrictions. However, the monolithic nature of all HDF5 variants introduces data access latency, whether on local drives or the commercial cloud. Because the hierarchical data structure must be flattened to be stored on disks, HDF5 readers must first read several byte blocks to decode the structure and find the address of the actual data in the file. Therefore, there are trade-offs in chunking the data: the smaller the chunk, the smaller the memory requirement, but the longer it takes to read the data schema map (Collette, 2013). Latency is made worse when compression is enabled. Another bottleneck of HDF5 is that the entire file has to be downloaded locally to be read. The downloading effectively duplicates data on local drives and is dramatically limited by the data volume (e.g., file size), the network connectivity or speed, and the storage availability on the local computing server. The computing overhead of copying large volumes of data is wasted if users only need a subset out of the whole data, which defeats the purpose of designing a single large file with small datasets within. Similar formats used in the geoscience community suffer the same issues on the cloud, such as NetCDF (Rew and Davis, 1990) and GeoTIFF (Ritter and Ruth, 1997).

Efforts have been made to optimize reading HDF5 on the cloud. One example is made by the Kerchunk project (see Data and Resources) to scan the HDF5 structure (byte ranges and compression information) and saves this information as index files to locate datasets directly without changing the original HDF5 file. In addition, the National Aeronautics and Space Administration (NASA)-supported sliderule project has developed a cloud-optimized read-only HDF5 library—H5-coro. H5-coro is currently used to read data from the ICESat-2 laser altimeter, stored directly as HDF5 files, into AWS S3 (Swinski *et al.*, 2023). Although both of these projects are under ongoing and rapid development, neither approach generalizes well for diverse H5 structures and data schema types (Ni, Swinski, and Denolle, 2023).

The ability to efficiently read small space–time subsets of data is essential for many scientific studies using DAS. This issue occurs with DAS, because, for example, the complexity of the cable geometry may challenge traditional seismic data processing. Rather than using the entire cable, DAS analysts may therefore select linear segments for surface-wave dispersion analysis (Fang *et al.*, 2023), velocity monitoring (Rodríguez Tribaldos and Ajo-Franklin, 2021), and beamforming (Nayak *et al.*, 2021). Furthermore, various factors affect the noise level of the DAS data, including the coupling between the cable and the ground and proximity to anthropogenic or natural noise sources. Finally, a cable may be long enough to span regions with different wave physics, for example, seafloor cable that extends onshore, offshore, and deep water regions (Tonegawa *et al.*, 2022). These effects naturally split the full cable into multiple segments that can be separately analyzed, which further necessitates a data-sharing system that favors such a reading pattern.

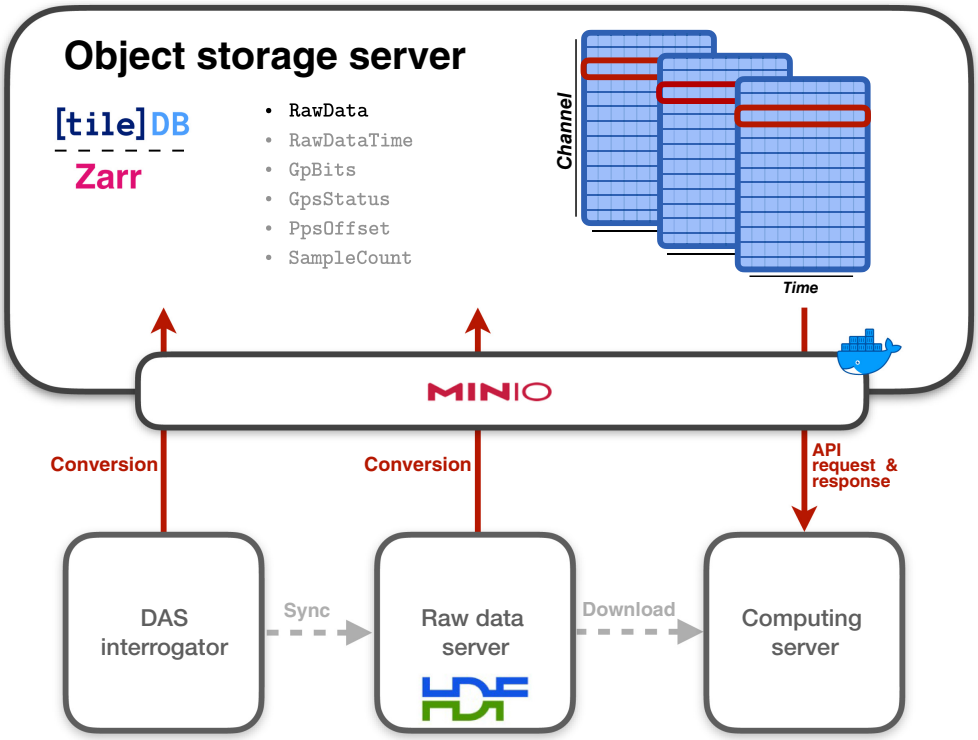


Figure 1. Workflow for DASstore: distributed acoustic sensing (DAS) data are converted into Zarr or TileDB format in the MinIO object storage server. The computing server sends a data request to the MinIO object storage and loads the data directly into the memory. A conventional workflow would be downloading the HDF5 file from the raw data server and loading data to memory on the server where computing is performed (gray dashed arrows). The data attributes converted from the raw HDF5 files are stored in the new format, which in the Ocean Observatories Initiative (OOI) examples are: RawData–2D DAS data, RawDataTime–1D time axis, with four customized datasets: GpBits, GpsStatus, PpsOffset, and SampleCount. These datasets are specific to the OptaSense interrogator of the OOI data. The color version of this figure is available only in the electronic edition.

DASstore: Object Storage and Format for DAS

This study proposes to develop and test an object storage framework for DAS; we deploy S3-compatible object storage locally and on AWS, and optimize data chunking for the performance of two cloud-optimized data formats—Zarr and TileDB (see Fig. 1).

Object storage for DAS

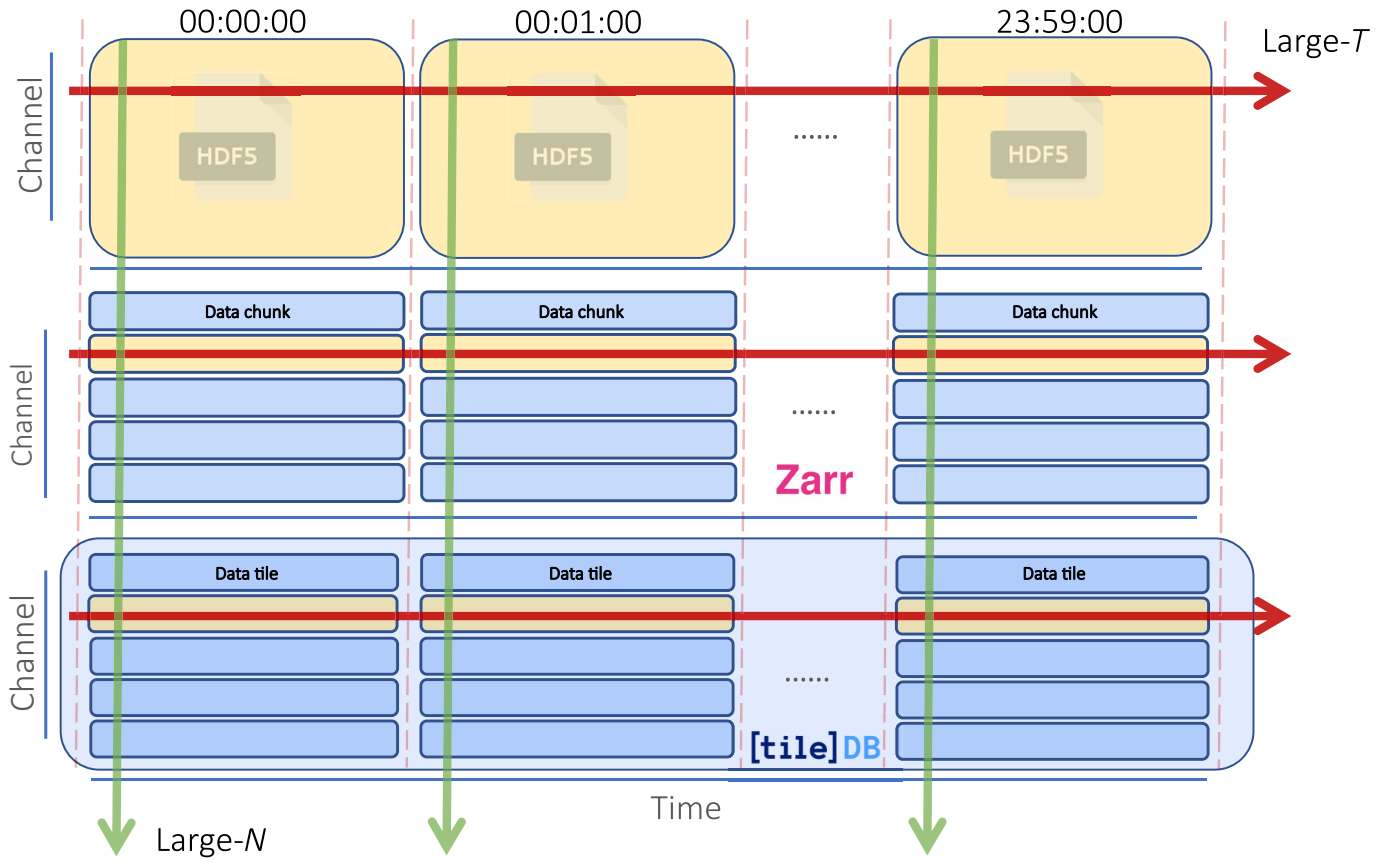
Object storage has seen increasing use as a framework for large geophysical archives. National archives of large datasets, such as data products from NASA, have migrated to cloud object storage to improve data access delivering over 100 PB on the cloud. The EarthScope Consortium Data Service (formerly IRIS-DMC and UNAVCO) is migrating to AWS S3 Cloud Storage and will deliver almost 2 PB of data upon their merger. Paired with object storage is the necessity for a cloud-optimized data format. Choosing a hierarchical data format such as HDF5 requires high-memory virtual machines and for users to load or download the entire HDF5 in memory locally. Although HDF5 was invented for better data sharing, it can dramatically slow down DAS data sharing and scientific investigation in modern distributed computing environments.

Here, we design a proof-of-concept implementation of a DAS data platform using cloud-optimized data structures and an object storage abstraction. We set up a server (either local or remote) with the MinIO object storage and convert raw DAS data into cloud-optimized formats. We find that this data platform outperforms a traditional, dedicated storage server (with a file system) for three reasons. First, we avoid having a local copy on the compute server and allow data chunks to be loaded directly into memory from the remote storage. Second, both cloud-optimized formats are highly compatible with the widely used Python data science and scientific computing environment, thereby enabling a user-friendly computing workflow with a shallow learning curve (e.g., NumPy, Dask, Xarray; Hoyer and Hamman, 2017). Third, the S3-compatibility of MinIO facilitates an effortless migration to the commercial cloud, for example, for heavier computation.

Data schema

Organizing data in a file is the first-order design choice for read and write performance. The chunking pattern is particularly determinant for formats like Zarr to perform well on object storage. A file may contain many data arrays representing different variables, which are split into chunks that form the minimum units to be read and processed in parallel. The size of a chunk is the minimum requirement for the memory of the computing unit. A small chunk will necessitate low memory, and a large chunk will necessitate a large memory. Thus, the number of processing units and memory allocation to each should be defined appropriately for the chunk size and file design.

Seismic data, in general, are multidimensional; it has a spatial dimension that may refer to the channel location and a



time dimension that refers to the time series that the instrument records. The schema of data containers for seismology is designed to organize the data according to either dimension (e.g., ASDF organizes the data as a dataset per channel). DAS data are, by nature of being constrained along the arc of a fiber, a 2D array with a time dimension and a channel dimension (i.e., arclength along the fiber). The HDF5 stored by the DAS interrogator has a simple data schema for which the RawData object is the 2D DAS data, and the RawDataTime object is the 1D time axis.

```
Acquisition
| – – Raw[0]
|      |– – RawData (47500, 12000), int32
|      |– – RawDataTime (12000,), int64
|
| – – Custom
       | – – GpBits (12000,), uint8
       | – – GpsStatus (12000,), uint8
       | – – PpsOffset (12000,), uint32
       | – – SampleCount (12000,), int64
```

Three strategies exist for chunking a 2D array with spatial and temporal dimensions. The first strategy aims to optimize long-time series data investigations (sometimes referred to by the community as "large-T," red arrow in Fig. 2): the optimal way would be to have chunks that are elongated on the time axis and narrow along the spatial axis. A second strategy would be to focus on spatial investigations (sometimes referred to by the community as "large-*N*," green arrow in Fig. 2): the optimal way would be to have chunks that are short on the time axis but long on the spatial axis. A third strategy would be to compromise between the first two.

Figure 2. Chunking data in the three formats. The raw data are stored from the interrogators as a single file for each minute of recording. The Zarr schema separates each HDF5 dataset into a number of chunks; each chunk is a single object. The TileDB schema collects the entire data into a single array; the chunks are "tiles" and are concatenated in time. The red arrow represents the case for a "large-T" analysis with a subarray of the DAS cable. The green arrow represented a "large-*N*" analysis with a smaller time window. The color version of this figure is available only in the electronic edition.

Formatting the metadata

There are three types of metadata in this context. The first is associated with each object in the object storage, which is used to identify objects and their properties. The storage server globally manages the object metadata and is part of the MinIO built-in design. The second is the format-specified metadata that is inherently part of the format, for example, the ARRAY_DIMENSIONS field in Zarr metadata to be compatible with Xarray. The third is the research-defined metadata that describes the DAS experiment. Here, we focus on the

TABLE 1
Linux Servers Used in the Tests

Server	Network	CPU	RAM	Main Storage
Raw data server	1 Gbps	96	754 GB	16 TB HDD
Object storage server	10 Gbps	16	92 GB	16 TB HDD
Compute server	10 Gbps	112	1007 GB	1 TB NVMe
Elastic compute cloud (m5.xlarge)	Up to 10 Gbps	4	15 GB	8 GB SSD

CPU, central processing unit.

convention proposed by the DAS Research Coordination Network (RCN)—a National Science Foundation-supported community effort to define common metadata for DAS. As of December 2022, the DAS-RCN defined five levels of metadata describing an experiment under this convention: Overview, Cable, Fiber, Interrogator, and Acquisition. These metadata are saved as key-value-pair attributes with the raw data. In addition, channel-specific information (channel number, location, and uncertainty) is saved in a separate comma-separated values file.

Performance Test on DASstore

We design performance tests based on seismological use cases to optimize the DAS data chunking schema for the best performance of the object storage. We consider data (1) converting data from HDF5 to Zarr or TileDB, and (2) implementing a feature extraction workflow. Three local servers are used for the benchmark tests: one raw data server storing HDF5 DAS data, one object storage server deployed in a single hard disk drive (not distributed), and one computing server as the client to request the data and perform computing. Table 1 lists the hardware specification of these servers relevant to the optimization criteria, including an AWS EC2 instance for the test on the cloud.

We conducted these tests using the Ocean Observatories Initiative (OOI) DAS experiment that was conducted in the first week of November 2021 on two ocean-bottom cables (north cable and south cable) of the OOI Regional Cable Array (Wilcock and the Ocean Observatories Initiative, 2023). We focus on the records of a 95 km long segment of the south cable, which goes from the land station westward until an optical repeater at a water depth of ~1500 m. The data were recorded by an OptaSense interrogator unit with a gauge length of 51.05 m, a channel spacing of 2.04 m, 47,500 channels, and a sampling rate of 200 Hz. The data are stored in HDF5 files every minute (~1.2 GB each), and 7.1 TB was collected. Almost 50,000 channels are rather large for typical DAS experiments (Spica *et al.*, 2023), and reading a single HDF5 file only gives 1 min of data and already requires large memory. To reconstitute longer time series requires selecting subarrays (groups of channels) within the file, which is computationally expensive using HDF5. This makes OOI DAS data ideal for testing our platform.

We design two performance tests to emulate potential use cases. First, a writing test evaluates the costs of converting the DAS HDF5 data into Zarr or TileDB, and whether it can be done in near-real time. Second, we conduct a reading test that evaluates the computational costs of the minimal seismological workflow that performs simple feature extraction (peak amplitude) after preprocessing (detrending, demeaning, and filtering). This second test aims to represent the use case of selecting a group of channels (e.g., a subarray of the DAS cable) and extracting features. More specifically, we extract 10 min subarray data with 5000 channels, and apply the typical processing steps of detrending and tapering to each channel, followed by a second-order [0.01, 1] Hz Butterworth band-pass filter and output the maximum amplitude of each channel. We perform these tests on our local Linux servers (see Table 1), emulating the case of a DAS experiment and archives being run by an individual investigator's computing system.

Baseline

We choose a baseline test that emulates how researchers traditionally access remote HDF5 data. We first download the OOI DAS data from the storage server to the computing server. The total downloading time depends mostly on the ethernet bandwidth between the two servers. Using a single process that saturates the full bandwidth, it takes 110 s to download all files (11.7 GB, 106.3 MB/s on average). The feature extraction workflow takes an additional 12.9 s on average with a single process. We also chunk the data into HDF5 files with varying chunk sizes (byte shuffle filter and LZ4 compressor enabled). Then, we download and run the feature extraction workflow separately on the compute node. This baseline test demonstrates that data download is a bottleneck and particularly inefficient when selecting several channels out of the array.

Writing test: data conversion costs

To test the I/O performance of the platform, we first test data conversion from the raw data server to the object storage. We sequentially convert 10 HDF5 files on the raw data server to the object storage in Zarr and TileDB formats. We use a single process to ensure that neither network nor disk I/O is saturated. The test demonstrates the feasibility of collecting and converting DAS data in real time. The test also represents the canonical workflow for users interested in earthquake waveforms: each HDF5 file is only 1 min of data, typically shorter than the seismic waveforms of moderate-to-large-size tectonic earthquakes. To measure the expense of converting and hosting DAS data in these formats, we track the conversion time, array size, and the number of objects after conversion. We do not test on AWS S3

TABLE 2
Ocean Observatories Initiative (OOI) Distributed Acoustic Sensing (DAS) Data Conversion from HDF5 into Zarr and TileDB Formats

	Zarr		TileDB	
Chunk or Tile Size	Writing Time (s)	Number of Objects	Writing Time (s)	Number of Objects
12000 × 1	1349.0	475,004	196.8	33
12000 × 2	894.1	237,504	193.2	33
12000 × 3	833.1	158,344	196.5	33
12000 × 4	695.4	118,754	197.3	33
12000 × 5	551.9	95,180	202.4	33
12000 × 10	460.8	47,504	–	–
12000 × 50	279.4	9,504	–	–
12000 × 100	326.3	4,754	–	–
12000 × 500	248.8	954	–	–
12000 × 1000	242.4	660	–	–

HDF5 files are converted directly to the object storage from the raw data server. The underlined numbers highlight the writing time shorter than the data duration, indicating the configurations that would allow real-time conversion.

servers to emulate the case of researcher-level data acquisition without edge computing capabilities.

We first test using different chunk-size configurations. We define a single Zarr array and concatenate DAS data along the time axis. To have a more straightforward data schema and have the Zarr compatible with more existing tools (e.g., Xarray), we group all datasets that are described in Figure 1 (10 min of data and all channels) and flatten the hierarchical structure of the HDF5 file. Zarr saves each chunk as an individual object (each 1 min long). For a fixed data size, choosing a chunk size determines the number of objects after conversion. The test begins with a small chunk (12,000 × 1, equivalent to 1 min of data at 200 Hz), and the conversion takes ~22 min with ~0.48 million objects (each a single channel for 1 min of recording or 25 KBs). Because we increase the chunk size to 1000 channels per chunk, each object size increases, the number of objects in the Zarr array decreases, and the writing time decreases. Table 2 demonstrates that the data conversion would become as short as the duration of the record (e.g., near-real time) for a small chunk size of five channels per object.

Selecting an optimal chunk size is critical for Zarr to perform well. A larger chunk size could bring more network overhead and slow down reading. On the other hand, small chunks have better reading performance, but many small objects could slow down the object storage as data grows. This latter latency may be due to our choice of hardware and may not exist on AWS S3. It is recommended to have at least a 1 MB chunk size, corresponding to grouping 50 channels in our case. The size of the Zarr array after conversion is almost equivalent to the original HDF5 files using a byte shuffle filter and LZ4 compressor. Thus, the conversion does not create more storage needs for the storage.

Next, we test the TileDB format using different tile size configurations. We concatenate DAS data along the time axis for a 10 min window and store it in a single TileDB array. Similar to the chunks in the Zarr format, a tile is the atomic unit of reading and writing for TileDB. This array has 10 tiles of 1 min data for each channel group. For the OOI DAS, we define the time dimension as an unsigned 32-bit integer (uint32), which allows $2^{32} - 1$ time index in the array (equivalent to ~248 days for a 200 Hz sampling rate). We use the same byte shuffle filter and LZ4 compressor as the test for Zarr when writing the data, which generates slightly bigger objects after conversion. As a result, TileDB generates only 33 objects for all tile size configurations: 10 objects correspond to 10-time data (called fragments, including data and data schema) and two additional objects per writing operation that store the changes and commits. In our case, we do not have data that gets updated over time and do not need to leverage the data versioning capabilities of TileDB. Unlike the Zarr format, the conversion keeps relatively steady at about 20 s as the tile size increases.

Reading test: seismological use case of feature extraction

We request a 5000-channel subarray (adjacent arrays in space and the data) on the computing server from the storage server to conduct the feature extraction workflow using the same 10-min OOI DAS data in Zarr and TileDB format. We detrend, taper, and [0.01, 1] Hz Butterworth band-pass filter each channel data. Here, we use Open-Message Passing Interface (MPI) to distribute the channel indexes to each process and parallelize the workflow using a different number of processes. The results are shown in Figure 3. Each run reads the same amount of data (5000 channels for 10 min) and is distributed over a variable number of processes.

The performance tests of Zarr indicate a sweet spot in terms of chunk size and the number of MPI processes for optimal read time. A small chunk (under 50 channels) requires large query times but decreases as a function of MPI processes. This may be attributed to the accumulated object storage latency, because a small chunk size corresponds to more S3 requests to get the same amount of data. On the contrary, a large chunk read is faster, but the network overhead and compute times increase with the number of MPI processes. Grouping more channels in a single object effectively increases the data granularity, potentially creating more network overhead, especially when a smaller subset of the data is requested (e.g., as processes increase). A similar pattern is visible on the AWS test (see the Testing the migration to commercial cloud section).

TileDB is fully parallelized internally. The format indexes the array and supports byte-range requests, and there is minimal

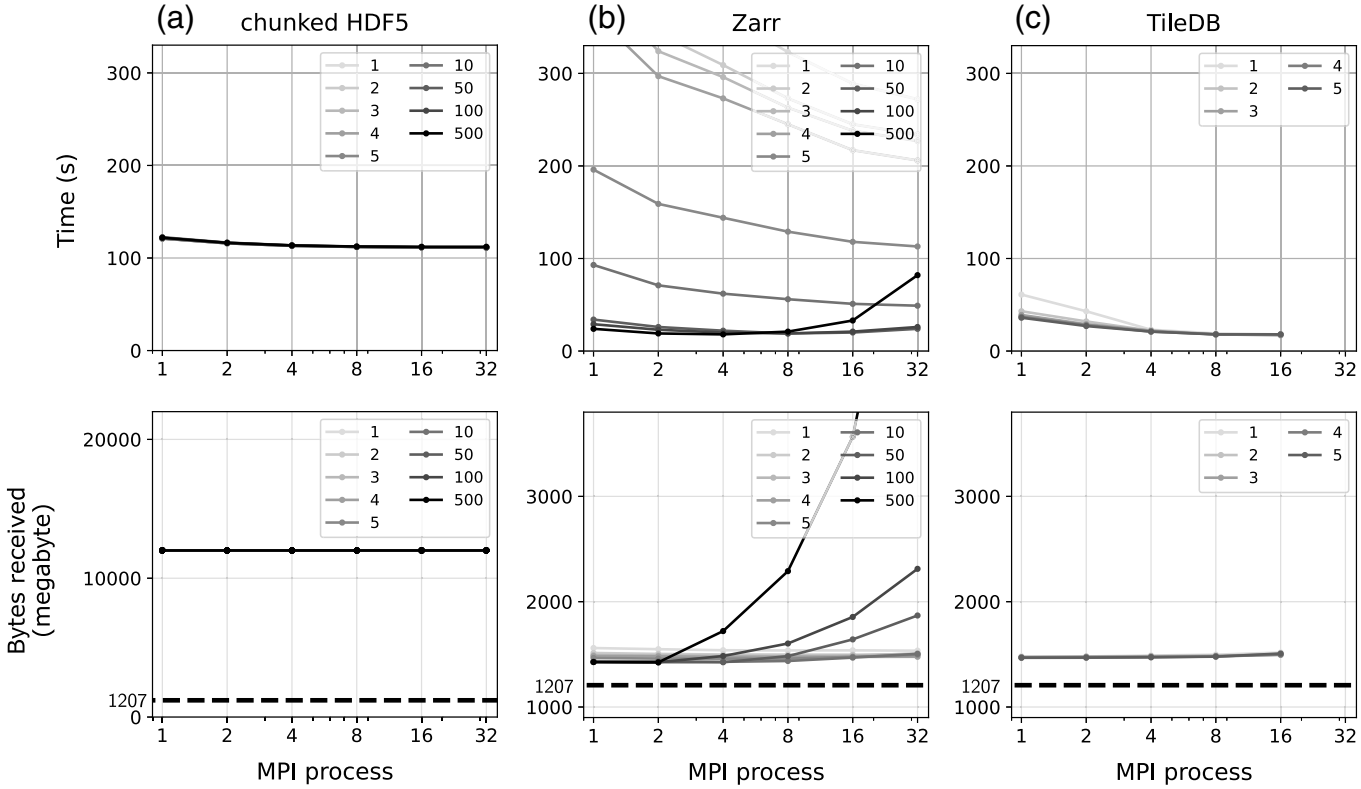


Figure 3. Results of the reading test for (a) chunked HDF5, (b) Zarr, and (c) TileDB format. For chunked HDF5, the compute server downloads the entire HDF5 file, reads a subarray of 5000 channels, and then performs the feature extraction on each channel. The compute server requests a subarray of 5000 channels from the object storage and performs the same feature extraction workflow. The upper panels show the time to finish the test, and the lower panels show the total bytes received by the compute server. The line color indicates the chunk or tile size for Zarr or TileDB arrays. All lines are overlapping with each other for the chunked HDF5 test. The black dashed line in the lower section shows the theoretical data size after compression.

extra network overhead. The performance with respect to tile sizes up to five channels was almost identical throughout the test.

Testing the migration to commercial cloud

We test our platform on AWS. We upload the OOI DAS data to AWS S3 and query data on an m5.xlarge EC2 instance (general purpose, 4 vCPU, 15 GB RAM) from the S3 bucket. We use 12,000 × 5 tile size for TileDB and 12,000 × 50 chunk size for Zarr, which we found optimal between read and write tests. Figure 4 shows the reading test implemented on AWS. The results are similar to those implemented through MinIO and match well with a strong scaling fitting (dotted line in Fig. 4a), assuming the network transmission time stays constant.

We identify three advantages of using the commercial cloud service to host the DAS data. First, there is usually no cost for the data transmitted into the cloud (no ingress cost), and storing data on the cloud for short periods of time is relatively cheap. For example, it costs $25 U.S. per month per TB on AWS S3; the storage costs would be less than $1 U.S. per day for a one-day experiment. Second, cloud storage services do not impose a space limit, which is different from local storage, for which the quotas are set by the hardware or the cluster centers. Therefore, cloud storage is an ideal repository for write-once-read-many scientific data. Third, commercial cloud storage has better back-end support and can handle a higher data request rate than the local object storage deployment (e.g., MinIO introduced in the previous section).

For most seismological applications, almost no network bottleneck exists between the computing instance and the object storage within the same AWS region, even on the lowest-tier virtual machine. If the computing is performed in the same region of the data, the data transmission is free. This is equivalent to having a massive file storage server with fast data query rate.

Application: SeaDAS-N Cross Correlation on AWS

We next carry out a standard ambient noise cross-correlation analysis using DAS data. We use the continuous records of an urban DAS experiment, SeaDAS-N, collected by the UW FiberLab between April 2022 and March 2023. The dark fiber used is owned by the University, and runs from the Atmospheric and Geophysics (ATG) building in the UW Seattle campus to the UW Bothell campus. The fiber runs mostly underground, but the cable segments are above ground,

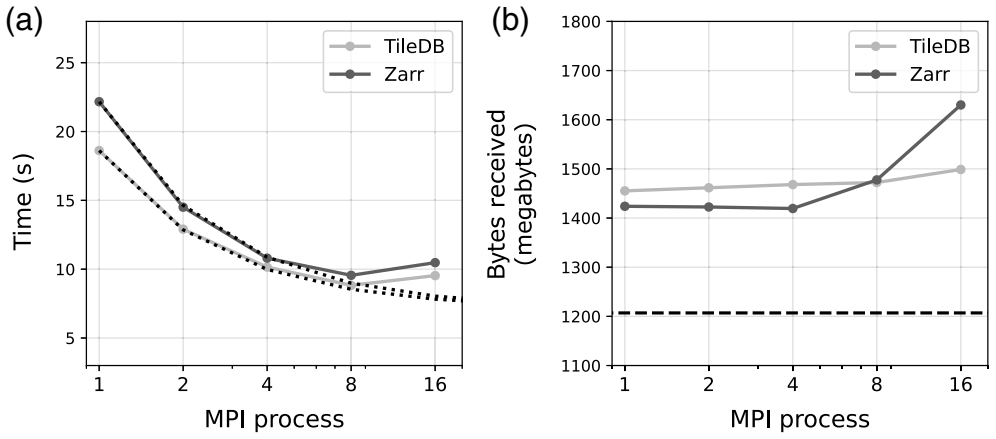


Figure 4. Results of the scaling reading test on Amazon Web Service (AWS). Panel (a) shows that the reading time averaged over 5 runs. The dotted line indicates a strong scaling fitting, assuming that the network transmission time stays constant. Panel (b) shows the total bytes received from the object storage averaged over five runs. The line color indicates the data format (Zarr and TileDB). The black dashed line in the right panel (b) shows the theoretical data size after compression.

which can be observed in the raw DAS amplitude and confirmed by independent distributed sensing measurements (temperature and strain sensing). The data were collected using a Sintela Onyx interrogator v.1.0 with 2089 channels (4.78 m channel spacing, 9.56 m gauge length) at a 100 Hz sampling rate. Figure 5 shows one-hour SeaDAS-N data as an example.

We select data from December 2022 and perform a typical workflow for ambient noise seismology applied to DAS data. This type of analysis is common for seismometers to construct surface waves and perform tomography (Shapiro *et al.*, 2005; Ritzwoller *et al.*, 2011; Lee *et al.*, 2014; Sager *et al.*, 2020; Cheng *et al.*, 2021; Zeng *et al.*, 2021; Yang, Atterholt, *et al.*, 2022) or allow for monitoring changes in the subsurface (Sens-Schönfelder and Wegler, 2006; Donaldson *et al.*, 2019; Rodríguez Tribaldos and Ajo-Franklin, 2021; Cheng *et al.*, 2022). For DAS data, most cross-correlation analyses have used a subset of the data (Tonegawa *et al.*, 2022; Viens *et al.*, 2023), and only a few examples exist today of full cable cross correlations (Rodríguez Tribaldos and Ajo-Franklin, 2021).

We test the seismological use case of ambient noise seismology with our data platform on AWS. One month of SeaDAS-N data stored on local drives is converted to Zarr and uploaded to an AWS S3 bucket in the us-west-2 region (Oregon). One minute of 100 channels is chunked into a single object on S3. It takes 5 hr to upload 1.7 TB SeaDAS-N data using four processes. A modified version of NoisePy (Jiang and Denolle, 2020) is containerized and submitted to the AWS Batch service for parallelization. Using all 2089 channels (~9.98 km), the correlation of one-month data is split into 248 small tasks. We distributed the tasks using AWS autoscaling: a service that automatically manages the type and number of EC2 instances based on the job requirements. Each task contains the following workflow: (1) requests 3 hr of SeaDAS-N data from S3, (2) processes the ambient field with one-bit temporal normalization and spectral whitening, (3) calculates the cross correlation, and (4) linearly stacks the cross correlations for each hour of data. In contrast to using HDF5 as the local intermediate data product as designed in NoisePy, each task writes the three hourly-stacked cross correlations directly to another S3 bucket in TileDB format. Figure 6 shows a daily stacking of the correlation function using channel 500 as the virtual source.

All tasks run on AWS memory-optimized (R-family) EC2s in the us-west-2 region (collocated with the data). The typical cost (r5.large; 2 vCPU; 15 GB RAM) is $0.0348 U.S. per hour on average, using preemptible SPOT instances (unused virtual machines sold at a much lower rate). On average, it takes 24 hr per task to compute 393 million cross correlations, with ~98 billion correlation operations performed in total. Because more tasks are completed, the autoscaling group automatically shuts down idling instances to optimize cost. Because the data I/O is fully implemented through our data platform, no extra local storage ("block storage") is required for each computing instance. In addition, there is no cost during data uploading and transmission within the cloud, but it costs on average $1.3 U.S. and $2.8 U.S. per day storing 1.7 TB DAS data and 3.7 TB correlation function on the cloud. Researchers may download the data product (egress cost applied) or conduct further analysis on the cloud. Overall, we spent ~$430 U.S. to cross correlate and download the stacked data. Although SPOT instances are preemptible, in our experience the small instances (r5.large) are rarely recalled. When they are recalled, batch automatically restarts new instances to relaunch the job that was interrupted. Furthermore, minimizing each job run on batch decreases the risk to lose intermediate results.

The cross correlation between channel 500 and all other channels of the DAS cable segment (see Fig. 5) is shown in Figure 6. We filter the data between 1 and 20 Hz. We find a typical cross-correlation image: waves propagate from the virtual source at about 400–500 m/s. The most likely wave type in this context is surface waves. Given the frequency band, it is not unusual to see slow-moving surface waves in shallow sediments, particularly in this area (Stephenson *et al.*, 2019). DAS correlations also often exhibit coherent, zero-lag signals that

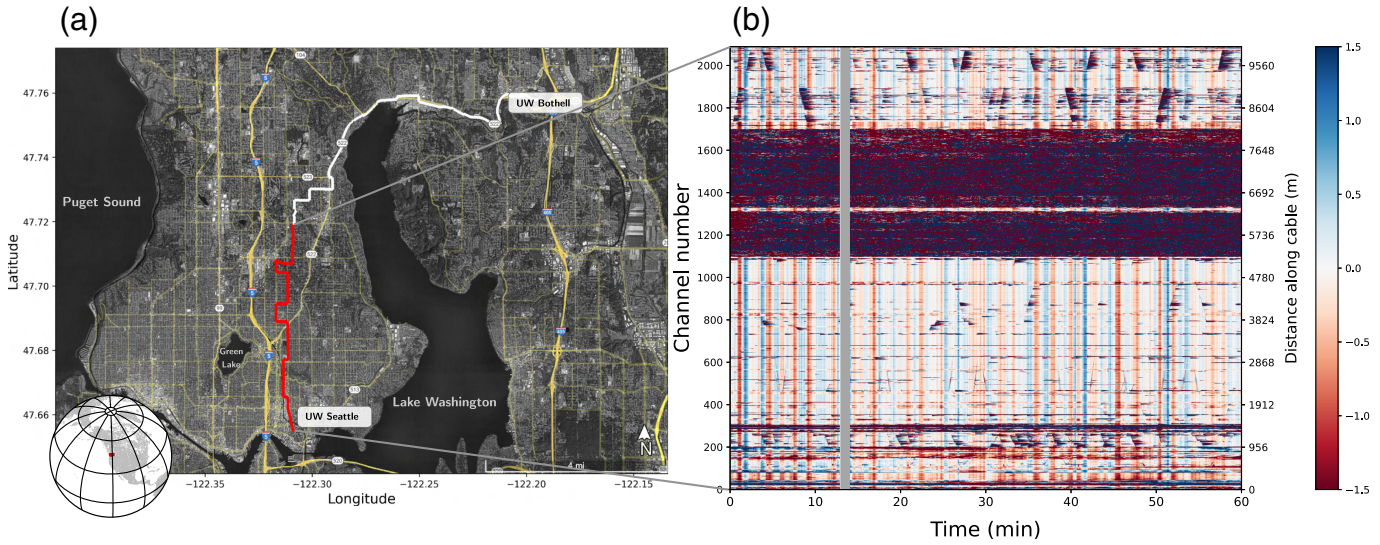


Figure 5. (a) Map showing the location of the SeaDAS-N cable (solid line) connecting Seattle and Bothell campus of the University of Washington, United States. Only the red segment was interrogated in December 2022 for data collection. The inset shows the location of the state of Washington in the North America. (b) One-hour SeaDAS-N raw data of all 2089 channels (in radians unit, a data volume of ~2.7 GB). The image starts 15 December 2022 at 13:30:00 UTC. The gray region indicates a one-minute data gap. The clipped data (channel ~1100–1700) are from the segment where the cable is above ground. The color version of this figure is available only in the electronic edition.

are instrumental and can be removed with an *f-k* filter (Tonegawa *et al.*, 2022). Parts of the cable are above ground (e.g., between channels 1100 and 1700), and therefore there is no correlation between channel 500 and these channels. Additional analysis of these data may involve a more careful analysis of the wave type (Rayleigh vs. Love waves, Fang *et al.*, 2023), the generation of dispersion curves, and shear-wave inversion for a velocity model.

Conclusions

This study presents the first cloud-native workflow for DAS research. We developed a cloud-optimized data platform using object storage and cloud-optimized data formats applicable to DAS seismological research. We attempted to represent a range of seismological use cases focusing on writing (I-input) tests for the archives and reading (O-output) tests for the users. Our benchmark tests demonstrate that one-time costs for data conversion from HDF5 (a common DAS data format) to Zarr or TileDB are manageable in near-real time.

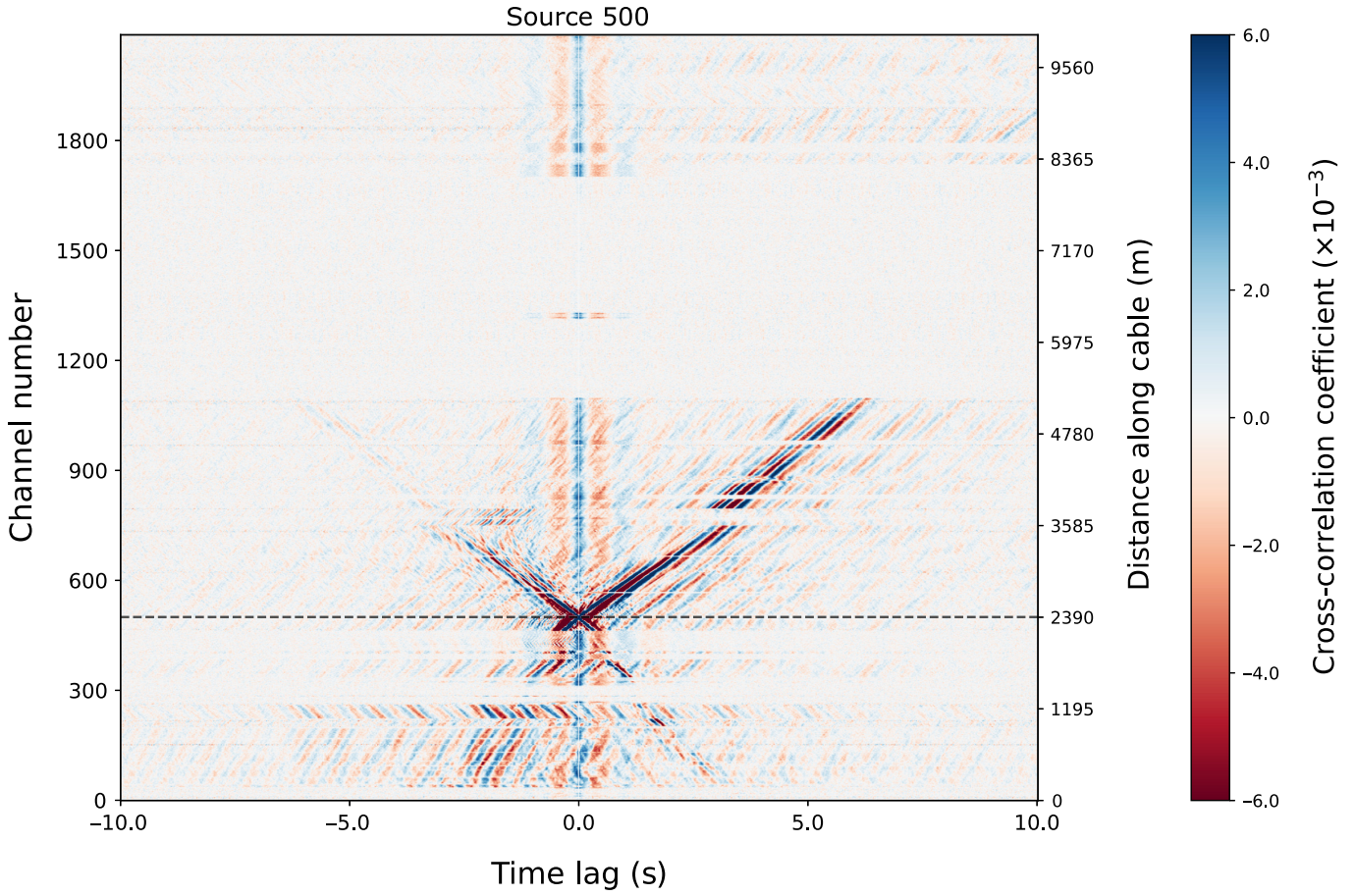
Data chunking is the first-order controlling factor in I/O performance. The I/O performance for Zarr indicates a sweet spot of 50–100 channels per chunk to be manageable for object storage for a typical 200 Hz sampling rate DAS experiment. Performance scaling in I/O is also favorable for Zarr and TileDB. TileDB natively parallelizes I/O. We distribute I/O using MPI for TileDB, and we demonstrate its good scaling performance up to 16 concurrent processes, after which latency becomes cumbersome, and our single disk object storage may not respond well under such concurrency. In whichever case, the performance has been dramatically improved. Researchers may find the optimal format and chunk or tile size that favors their data through a similar I/O benchmark and back-end specification.

This study focuses on performance using cloud services. We anticipate that users will seek to lower costs by minimizing compute time, memory requirements, and number of vCPU in the thread. However, long-term storage of 100 TBs data sets on the cloud is still cost prohibitive, there we have demonstrated a way for individual researchers to archive DAS data. We demonstrated an example using ambient noise cross correlations. We developed the workflow using an existing Python package (NoisePy) that we modified by moving intermediate data products from local storage (i.e., the most common workflow) to cloud storage, which greatly improved the performance.

The integration of DASstore with other DAS community tools in Python can be straightforward: DASstore is only a package for data query, and data are returned as NumPy arrays—a generic type for Python processing. We will use GitHub as the conduit for collaboration to initiate issues or discussions, and to support new cloud-native projects.

Data and Resources

The scripts of benchmark workflows are available through distributed acoustic sensing (DASstore's) GitHub repository (https://github.com/niyiyu/DASstore, Ni, Ragland, and Fatland, 2023). Chengxin Jiang adapted NoisePy to NoisePy4DAS, which we modified for cloud environments. The scripts to run the cross correlation on Amazon Web Services (AWS) and instructions on how to run AWS Batch using



autoscaling are available at https://github.com/niyiyu/NoisePy4DAS-SeaDAS. The Ocean Observatories Initiative (OOI) DAS data are available at http://piweb.ooirsn.uw.edu. The SeaDAS-N data from December 2022 are freely accessible (Lipovsky, 2023b) through our Python application programming interface (API) using the endpoint (https://dasway.ess.washington.edu), for which a tutorial can be found on the project repository page. The Kerchunk project is hosted at https://github.com/fsspec/kerchunk. DAS Research Coordination Network (RCN) is hosted at https://www.iris.edu/hq/initiatives/das_rcn. The information about the Southern California Earthquake Data Center (SCEDC) data on the AWS is available at https://registry.opendata.aws/southern-california-earthquakes/. The information about PoroTomo data on the AWS is available at https://registry.opendata.aws/nrel-pds-porotomo/. All websites were last accessed in October 2023.

Acknowledgments

This work is supported by eScience 2023 winter incubator project and the Seismic Computational Platform for Empowering Discovery (SCOPED) project under the National Science Foundation (NSF) Award Number OAC-2103701. The FiberLab was supported by the Murdock Charitable Trust. Results presented in this article were obtained using CloudBank (Norman *et al.*, 2021), which is supported by the NSF under Award Number 1925001. The authors also thank two anonymous reviewers for their thoughtful comments and suggestions, which greatly improved the article.

Figure 6. An example of ambient noise cross-correlation functions of SeaDAS-N using channel 500 as the virtual source. The channel number is labeled on the left *y* axis, and the distance along the cable is labeled on the right *y* axis. The black dashed line indicates the location of the virtual source. The color version of this figure is available only in the electronic edition.

References

Arrowsmith, S. J., D. T. Trugman, J. MacCarthy, K. J. Bergen, D. Lumley, and M. B. Magnani (2022). Big data seismology, *Rev. Geophys.* **60,** no. 2, e2021RG000769, doi: 10.1029/2021RG000769.

Atterholt, J., Z. Zhan, and Y. Yang (2022). Fault zone imaging with distributed acoustic sensing: Body-to-surface wave scattering, *J. Geophys. Res.* **127,** no. 11, doi: e2022JB025052, doi: 10.1029/2022JB025052.

Chambers, D., E. Martin, G. Jin, and A. H. S. Issah (2022). Dasdae/dascore: v0.0.7, *Zenodo*, doi: 10.5281/zenodo.7373559.

Cheng, F., B. Chi, N. J. Lindsey, T. C. Dawe, and J. B. Ajo-Franklin (2021). Utilizing distributed acoustic sensing and ocean bottom fiber optic cables for submarine structural characterization, *Sci. Rep.* **11,** no. 1, 5613.

Cheng, F., N. J. Lindsey, V. Sobolevskaia, S. Dou, B. Freifeld, T. Wood, S. R. James, A. M. Wagner, and J. B. Ajo-Franklin (2022). Watching the cryosphere thaw: Seismic monitoring of permafrost degradation using distributed acoustic sensing during a controlled heating

experiment, *Geophys. Res. Lett.* **49,** no. 10, e2021GL097195, doi: 10.1029/2021GL097195.

Collette, A. (2013). *Python and HDF5: Unlocking Scientific Data*, O'Reilly Media, Inc, Sebastopol, California.

Donaldson, C., T. Winder, C. Caudron, and R. S. White (2019). Crustal seismic velocity responds to a magmatic intrusion and seasonal loading in Iceland's northern volcanic zone, *Sci. Adv.* **5,** no. 11, eaax6642, doi: 10.1126/sciadv.aax6642.

Dong, B., V. R. Tribaldos, X. Xing, S. Byna, J. Ajo-Franklin, and K. Wu (2020). Dassa: Parallel das data storage and analysis for subsurface event detection, *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, IEEE, 254–263.

Douglass, A. S., S. Abadi, and B. P. Lipovsky (2023). Distributed acoustic sensing for detecting near surface hydroacoustic signals, *JASA Express Lett.* **3,** no. 6, doi: 10.1121/10.0019703.

Fang, J., Y. Yang, Z. Shen, E. Biondi, X. Wang, E. F. Williams, M. W. Becker, D. Eslamian, and Z. Zhan (2023). Directional sensitivity of das and its effect on Rayleigh-wave tomography: A case study in Oxnard, California, *Seismol. Res. Lett.* **94,** no. 2A, 887–897.

Feigl, K., N. Taverna, and M. Rossol (2016). Porotomo natural laboratory horizontal and vertical distributed acoustic sensing data, doi: 10.15121/1778858.

Hauksson, E., C. Yoon, E. Yu, J. R. Andrews, M. Alvarez, R. Bhadha, and V. Thomas (2020). Caltech/USGS southern California seismic network (SCSN) and southern California earthquake data center (SCEDC): Data availability for the 2019 Ridgecrest sequence, *Seismol. Res. Lett.* **91,** no. 4, 1961–1970.

Hoyer, S., and J. Hamman (2017). xarray: N-D labeled arrays and datasets in Python, *J. Open Res. Softw.* **5,** no. 1, 10.

Jiang, C., and M. A. Denolle (2020). Noisepy: A new high-performance python tool for ambient-noise seismology, *Seismol. Res. Lett.* **91,** no. 3, 1853–1866.

Kaneko, Y., and P. Shearer (2015). Variability of seismic source spectra, estimated stress drop, and radiated energy, derived from cohesive-zone models of symmetrical and asymmetrical circular and elliptical ruptures, *J. Geophys. Res.* **120,** no. 2, 1053–1079.

Krischer, L., J. Smith, W. Lei, M. Lefebvre, Y. Ruan, E. S. de Andrade, N. Podhorszki, E. Bozdağ, and J. Tromp (2016). An adaptable seismic data format, *Geophys. J. Int.* **207,** no. 2, 1003–1011, doi: 10.1093/gji/ggw319.

Lee, E.-J., P. Chen, T. H. Jordan, P. B. Maechling, M. A. Denolle, and G. C. Beroza (2014). Full-3-d tomography for crustal structure in southern California based on the scattering-integral and the adjoint-wavefield methods, *J. Geophys. Res.* **119,** no. 8, 6421–6451.

Lindsey, N. J., T. C. Dawe, and J. B. Ajo-Franklin (2019). Illuminating seafloor faults and ocean dynamics with dark fiber distributed acoustic sensing, *Science* **366,** no. 6469, 1103–1107.

Lindsey, N. J., S. Yuan, A. Lellouch, L. Gualtieri, T. Lecocq, and B. Biondi (2020). City-scale dark fiber DAS measurements of infrastructure use during the COVID-19 pandemic, *Geophys. Res. Lett.* **47,** no. 16, e2020GL089931, doi: 10.1029/2020GL089931.

Lipovsky, B. (2023a). Distributed Acoustic Sensing experiment in Whidbey Island, Washington, USA, doi: 10.6069/2GNS-7P48.

Lipovsky, B. (2023b). SeaDAS-N: Distributed Acoustic Sensing experiment in Seattle, Washington, USA, doi: 10.6069/R22Y-RQ65.

MacCarthy, J., O. Marcillo, and C. Trabant (2020). Seismology in the cloud: A new streaming workflow, *Seismol. Res. Lett.* **91,** no. 3, 1804–1812.

Miles, A., J. Kirkham, M. Durant, J. Bourbeau, T. Onalan, J. Hamman, Z. Patel, M. Rocklin, R. Dussin, and V. Schut (2020). zarr-developers/zarr-python: v2.4.0, doi: 10.5281/zenodo.3773450.

Nayak, A., J. Ajo-Franklin, and I. V. D. F. Team (2021). Distributed acoustic sensing using dark fiber for array detection of regional earthquakes, *Seismol. Res. Lett.* **92,** no. 4, 2441–2452.

Ni, Y., J. Ragland, and R. Fatland (2023). niyiyu/dasstore: DASstore, doi: 10.5281/zenodo.7826945.

Ni, Y., J.-P. A. Swinski, and M. Denolle (2023). Cloud-optimized ASDF-H5 for Seismology, doi: 10.22541/essoar.168298674.44060085/v1.

Norman, M., V. Kellen, S. Smallen, B. DeMeulle, S. Strande, E. Lazowska, N. Alterman, R. Fatland, S. Stone, A. Tan, *et al.* (2021). Cloudbank: Managed services to simplify cloud access for computer science research and education, *Practice and Experience in Advanced Research Computing*, Boston, Massachusetts, 18–22 July 2021, 1–4.

Papadopoulos, S., K. Datta, S. Madden, and T. Mattson (2016). The tiledb array data storage manager, *Proc. VLDB Endow.* **10,** no. 4, 349–360, doi: 10.14778/3025111.3025117.

Quinteros, J., J. A. Carter, J. Schaeffer, C. Trabant, and H. A. Pedersen (2021). Exploring approaches for large data in seismology: User and data repository perspectives, *Seismol. Res. Lett.* **92,** no. 3, 1531–1540.

Rew, R., and G. Davis (1990). Netcdf: An interface for scientific data access, *IEEE Comput. Graph. Appl.* **10,** no. 4, 76–82.

Ritter, N., and M. Ruth (1997). The geotiff data interchange standard for raster geographic images, *Int. J. Remote Sens.* **18,** no. 7, 1637–1647.

Ritzwoller, M. H., F.-C. Lin, and W. Shen (2011). Ambient noise tomography with a large seismic array, *C. R. Geosci.* **343,** nos. 8/9, 558–570.

Rodríguez Tribaldos, V., and J. B. Ajo-Franklin (2021). Aquifer monitoring using ambient seismic noise recorded with distributed acoustic sensing (DAS) deployed on dark fiber, *J. Geophys. Res.* **126,** no. 4, e2020JB021004, doi: 10.1029/2020JB021004.

Sager, K., C. Boehm, L. Ermert, L. Krischer, and A. Fichtner (2020). Global-scale full-waveform ambient noise inversion, *J. Geophys. Res.* **125,** no. 4, e2019JB018644, doi: 10.1029/2019JB018644.

Sens-Schönfelder, C., and U. Wegler (2006). Passive image interferometry and seasonal variations of seismic velocities at Merapi volcano, Indonesia, *Geophys. Res. Lett.* **33,** no. 21, doi: 10.1029/2006GL027797.

Shapiro, N. M., M. Campillo, L. Stehly, and M. H. Ritzwoller (2005). High-resolution surface-wave tomography from ambient seismic noise, *Science* **307,** no. 5715, 1615–1618.

Shearer, P. M. (2019). *Introduction to Seismology*, Cambridge University Press, New York.

Spica, Z. J., J. Ajo-Franklin, G. C. Beroza, B. Biondi, F. Cheng, B. Gaite, B. Luo, E. Martin, J. Shen, C. Thurber, *et al.* (2023). Pubdas: A public distributed acoustic sensing datasets repository for geosciences, *Seismol. Res. Lett.* **94,** no. 2A, 983–998.

Spica, Z. J., M. Perton, E. R. Martin, G. C. Beroza, and B. Biondi (2020). Urban seismic site characterization by fiber-optic seismology, *J. Geophys. Res.* **125,** no. 3, e2019JB018656, doi: 10.1029/2019JB018656.

Stephenson, W. J., M. W. Asten, J. K. Odum, and A. D. Frankel (2019). Shear-wave velocity in the Seattle basin to 2 km depth characterized with the krspac microtremor array method: Insights for urban basin-scale imaging, *Seismol. Res. Lett.* **90,** no. 3, 1230–1242.

Swinski, J., E. Lidwa, T. Sutterley, S. Henderson, D. Shean, C. E. Ugarte, J. J. Gearon, J. H. Kennedy, and P. Bot (2023). ICESat2-SlideRule/sliderule, doi: 10.5281/zenodo.7838015.

Tonegawa, T., E. Araki, H. Matsumoto, T. Kimura, K. Obana, G. Fujie, R. Arai, K. Shiraishi, M. Nakano, Y. Nakamura, *et al.* (2022). Extraction of p wave from ambient seafloor noise observed by distributed acoustic sensing, *Geophys. Res. Lett.* **49,** no. 4, e2022GL098162, doi: 10.1029/2022GL098162.

Viens, L., M. Perton, Z. J. Spica, K. Nishida, T. Yamada, and M. Shinohara (2023). Understanding surface wave modal content for high-resolution imaging of submarine sediments with distributed acoustic sensing, *Geophys. J. Int.* **232,** no. 3, 1668–1683.

Waagaard, O. H., E. Rønnekleiv, A. Haukanes, F. Stabo-Eeg, D. Thingbø, S. Forbord, S. E. Aasen, and J. K. Brenne (2021). Real-time low noise distributed acoustic sensing in 171 km low loss fiber, *OSA Contin.* **4,** no. 2, 688–701.

Wang, X., Z. Zhan, E. F. Williams, M. G. Herráez, H. F. Martins, and M. Karrenbach (2021). Ground vibrations recorded by fiber-optic cables reveal traffic response to covid-19 lockdown measures in Pasadena, California, *Commun. Earth Environ.* **2,** no. 1, 160.

Weers, J., A. Anderson, and N. Taverna (2022). The geothermal data repository: Ten years of supporting the geothermal industry with open access to geothermal data, *Tech. Rept.* National Renewable Energy Lab.(NREL), Golden, Colorado, available at https://www.nrel.gov/docs/fy22osti/82837.pdf (last accessed October 2023).

White, M. C., Z. Zhang, T. Bai, H. Qiu, H. Chang, and N. Nakata (2023). Hdf5eis: A storage and input/output solution for big multi-dimensional time series data from environmental sensors, *Geophysics* **88,** no. 3, F29–F38.

Wilcock, W., and the Ocean Observatories Initiative (2023). Rapid: A community test of distributed acoustic sensing on the ocean observatories initiative regional cabled array, doi: 10.58046/5J60-FJ89.

Wilcock, W. S., S. Abadi, and B. P. Lipovsky (2023). Distributed acoustic sensing recordings of low-frequency whale calls and ship noise offshore central Oregon, *JASA Express Lett.* **3,** no. 2, 026002.

Williams, E. F., Z. Zhan, H. F. Martins, M. R. Fernández-Ruiz, S. Martn-López, M. González-Herráez, and J. Callies (2022). Surface gravity wave interferometry and ocean current monitoring with ocean-bottom das, *J. Geophys. Res.* **127,** no. 5, e2021JC018375, doi: 10.1029/2021JC018375.

Yang, Y., J. W. Atterholt, Z. Shen, J. B. Muir, E. F. Williams, and Z. Zhan (2022). Sub-kilometer correlation between near-surface structure and ground motion measured with distributed acoustic sensing, *Geophys. Res. Lett.* **49,** no. 1, e2021GL096503, doi: 10.1029/2021GL096503.

Yang, Y., Z. Zhan, Z. Shen, and J. Atterholt (2022). Fault zone imaging with distributed acoustic sensing: Surface-to-surface wave scattering, *J. Geophys. Res.* **127,** no. 6, e2022JB024329, doi: 10.1029/2022JB024329.

Yu, E., A. Bhaskaran, S.-L. Chen, Z. E. Ross, E. Hauksson, and R. W. Clayton (2021). Southern California earthquake data now available in the AWS cloud, *Seismol. Res. Lett.* **92,** no. 5, 3238–3247.

Zeng, X., C. H. Thurber, H. F. Wang, D. Fratta, and K. L. Feigl (2021). High-resolution shallow structure at brady hot springs using ambient noise tomography (ANT) on a trenched distributed acoustic sensing (DAS) array, in *Distributed Acoustic Sensing in Geophysics: Methods and Applications*, Y. Li, M. Karrenbach, and J. B. Ajo-Franklin (Editors), American Geophysical Union, Hoboken, New Jersey, 101–110.

Manuscript received 5 June 2023
Published online 20 October 2023