Geophysical Journal International

Geophys. J. Int. (2025) **243,** 1–15
Advance Access publication 2025 August 18
Review Paper

https://doi.org/10.1093/gji/ggaf322

A review of cloud computing and storage in seismology

Yiyu Ni [1], Marine A. Denolle [1], Jannes Münchmeyer [2], Yinzhi Wang [3], Kuan-Fu Feng,[1,4] Carlos Garcia Jurado Suarez,[5] Amanda M. Thomas,[6] Chad Trabant,[7] Alex Hamilton[7] and David Mencin[7]

[1] *Department of Earth and Space Sciences, University of Washington,* 400015*th Ave NE, Seattle, WA* 98195 *, USA. E-mail: mdenolle@uw.edu*
[2] *Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, IRD, Univ. Gustave Eiffel, ISTerre, CS* 4070038058 *Grenoble Cedex* 9*, France*
[3] *Texas Advanced Computing Center, University of Texas,* 10100 *Burnet Rd, Austin, TX* 78758*, USA*
[4] *Department of Geology and Geophysics, University of Utah,* 115 *S* 1460 *E, Salt Lake City, UT* 84112*, USA*
[5] *eScience Institute, University of Washington, Campus Box* 351570*,* 391015*th Ave NE, Seattle, WA* 98195*, USA*
[6] *Department of Earth and Planetary Sciences, University of California, One Shields Avenue, Davis, CA* 95616*, USA*
[7] *EarthScope Consortium,* 1200 *New York Avenue NW, Suite* 400*, Washington DC* 20005*, USA*

Accepted 2025 August 14. Received 2025 August 13; in original form 2025 May 23

SUMMARY

Seismology has entered the petabyte era, driven by decades of continuous recordings of broad-band networks, the increase in nodal seismic experiments and the recent emergence of distributed acoustic sensing (DAS). This review explains how cloud platforms, by providing object storage, elastic compute and managed data bases, enable researchers to 'bring the code to the data,' thereby providing a scalable option to overcome traditional HPC solutions' bandwidth and capacity limitations. After literature reviews of cloud concepts and their research applications in seismology, we illustrate the capacities of cloud-native workflows using two canonical end-to-end demonstrations: (1) ambient noise seismology that calculates cross-correlation functions at scale, and (2) earthquake detection and phase picking. Both workflows utilize Amazon Web Services, a commercial cloud platform for streaming I/O and provenance, demonstrating that cloud throughput can rival on-premises HPC at comparable costs, scanning 100 TBs to 1.3 PBs of seismic data in a few hours or days of processing. The review also discusses research and education initiatives, the reproducibility benefits of containers and cost pitfalls (e.g. egress, I/O fees) of energy-intensive seismological research computing. While designing cloud pipelines remains non-trivial, partnerships with research software engineers enable converting domain code into scalable, automated and environmentally conscious solutions for next-generation seismology. We also outline where cloud resources fall short of specialized HPC—most notably for tightly coupled petascale simulations and long-term, PB-scale archives—so that practitioners can make informed, cost-effective choices.

Key words: Computational seismology; Machine learning; Seismic interferometry.

1 INTRODUCTION

Seismology has entered a 'petabyte era,' where seismic networks and nodal array experiments routinely generate more data than traditional workstations and institutional clusters can store or analyse. Data from more than 70 000 seismometers has surpassed 1 PB on the EarthScope Data Archive (Arrowsmith *et al.* 2022). Novel technologies, such as Distributed Acoustic Sensing (DAS), which collect thousands of sensors per experiment, are also increasing the data storage needs and already surpassing the PBs of data (i.e. data sets exceeding 10^{15} bytes) collected and shared (Zhan 2020; Spica *et al.* 2023; Wuestefeld *et al.* 2024). State-of-the-art research networks and transfer tools now enable the movement of a petabyte of scientific data in about a day under optimal conditions (e.g. dedicated inter-institutional research networks with high-throughput protocols such as Globus and dCache), but remain challenging to move through normal internet bandwidth. Moreover, seismic data services standards well-adopted by data centres are not well suited for big-data seismology studies (Quinteros *et al.* 2021a; Arrowsmith *et al.* 2022). While seismologists' workflows traditionally involve analysing data by downloading from archives and working on-premise, this model faces significant difficulties with analysis that requires more than several TBs of data. Researchers are exploring cloud computing to address these bottlenecks, which brings code to data and offers scalable storage and processing.

To turn this data deluge into discovery, seismologists increasingly look to cloud platforms that put compute next to the archive. In this review, we use *cloud computing and storage* to mean shared remote computing and data centres run by global firms like Amazon Web Services (AWS), Google Cloud Platform (GCP) and Microsoft Azure, as well as European research clouds such as SURF and European Grid Infrastructure (EGI) that let geophysicists borrow computing clusters and storage on demand, pay only for the time used. Commercialized cloud computing emerged in the early 2000s, when the industry began providing capabilities such as large-scale object storage and on-demand computing. Geophysicists have traditionally used high-performance computing (HPC) centres to deliver tightly coupled, job-scheduled computing on a shared filesystem, often driven by and designed for large-scale numerical simulations such as wavefield simulations (e.g. SPECFEM; Peter *et al.* 2011) and earthquake dynamics (e.g. SeisSol; Heinecke *et al.* 2014). In contrast to HPC systems, cloud providers deliver elastic resources as metred services that users spin up and pay for only when needed. This encompasses several key design elements of their inner working, including virtual machines (VMs), cloud storage and advanced services like data bases and serverless services, which we introduce in this review. Practically, this means a researcher can open a web portal and launch virtual machines or clusters in several minutes. The cloud's hidden orchestration layer juggles requests by slicing real hardware into many small 'virtual' ones, which gives users elastic capacity but also means users share logically isolated hardware with others. That convenience translates into strengths (instant scale-up, pay-as-you-go, global access) and weaknesses (performance can vary, data may cross the public internet, long-term storage fees add up) that we further discuss. Harnessing cloud infrastructure could fundamentally change how seismologists handle big data, making analyses faster and more collaborative.

Cloud offers tremendous opportunities for easy access to object storage, a centralized, affordable and widely accessible solution for massive data archives. In contrast to HPC, object storage enables hosting and sharing PBs of public geoscientific data (Zhuang *et al.* 2020; Abernathey *et al.* 2021; Gentemann *et al.* 2021). While individual data queries may have modest throughput speeds (10–1000 MB s^{-1}), the large parallelization capabilities of cloud systems allow throughput speeds comparable to those of HPC scratch systems (10–100 GB s^{-1}). Moreover, cloud providers are storing PBs of publicly available and free-access geoscientific data (Abernathey *et al.* 2021). In seismology, hosting seismic data on the cloud is rising, as we illustrate in Fig. 1. The Southern California Earthquake Data Center (SCEDC) was the first to provide archives of regional seismic networks as open data sets on the commercial cloud (Yu *et al.* 2021). It was recently followed by the Northern California Earthquake Data Center (NCEDC) and the EarthScope-operated Seismological Facility for the Advancement of Geoscience repository (formerly Incorporated Research Institutions for Seismology Data Management Center, IRIS DMC). Cloud has also been a promising storage solution for DAS data [e.g. PoroTomo, https://registry.opendata.aws/nrel-pds-porotomo/ (Feigl 1969), and Ridgecrest DAS (Yu *et al.* 2021)].

A typical entry point for scientists to cloud computing is through a freely accessible JupyterHub with a backend running on cloud platforms. Open-access JupyterHubs with notebooks (e.g. Binder, Google Colab and EarthScope GeoLab) lower the entry barrier to the cloud by giving users a ready-made Python environment. Users can automate software-to-infrastructure, using tools such as `repos2docker` to containerize software and automatically provide

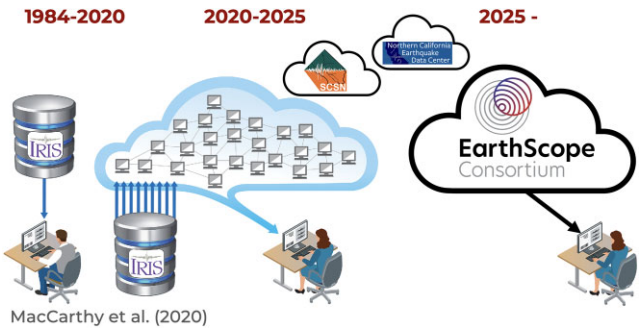

Figure 1. Evolution of cloud computing in seismology: before 2020, most computational workflows involved downloading data from the Incorporated Research Institutions for Seismology Data Management Center (IRIS DMC) and other data centres and working locally. Between 2020 and 2025, seismologists have investigated the use of elastic computing by pulling data from existing archives and processing directly on the cloud (e.g. MacCarthy *et al.* 2020). At the same time, two regional seismic networks copied their archives of earthquake catalogues and seismic waveforms to Amazon Web Services (AWS): the Southern California Seismic Network (SCSN) and the Northern California Seismic Network (NCSN). Since 2025, the EarthScope Consortium has migrated its petabyte-scale archive to the cloud, enabling researchers to pull and compute directly on the cloud.

access to cloud-hosted virtual machines. In particular for seismologists, Krischer *et al.* (2018) has pioneered the use of cloud-hosted Jupyter notebooks, utilizing modest yet free Binder instances provided and donated by various cloud providers (e.g. OVHcloud as of 2025). Alternatively, Google Colab provides free access to modest-sized virtual machines, which utilize GCP resources with pre-defined Python environments. Cloud is also an on-demand platform to host educational materials (Denolle *et al.* 2025).

Considering these basics, this article reviews how cloud computing has been applied in seismological research. We discuss storage solutions for seismic data (Section 2), data bases (Section 3), the various types of computing resources (Section 4), showcase experiments conducted at scale on large archives of broad-band seismic data (Section 5), and cloud-enabled visualizations (Section 6). We also present a series of experiments that have targeted archetypes in big-data seismology: (1) data mining using deep learning models to detect seismic events, and (2) ambient field seismology that requires intensive generation of cross-correlation at scale. Both tasks are characterized by a high data intake and large computational requirements, yet differ in output and processing specifics. In particular, we focus on the specific requirements of each workflow and how they affect the choice of cloud tools. Finally, we present our recent experience in running a cloud seismology workshop (Section 7) and discuss the opportunities, cost and challenges in cloud computing for seismological research (Section 8).

2 CLOUD STORAGE

2.1 Cloud-hosted data archives

Transferring and sharing data between institutions is essential for large-scale collaborative research in seismology. Software from command-line tools (e.g. `sftp`, `wget`) to hosted services (e.g. Globus; Allen *et al.* 2012) is made available to securely and efficiently share data over the internet to facilitate the transfer of

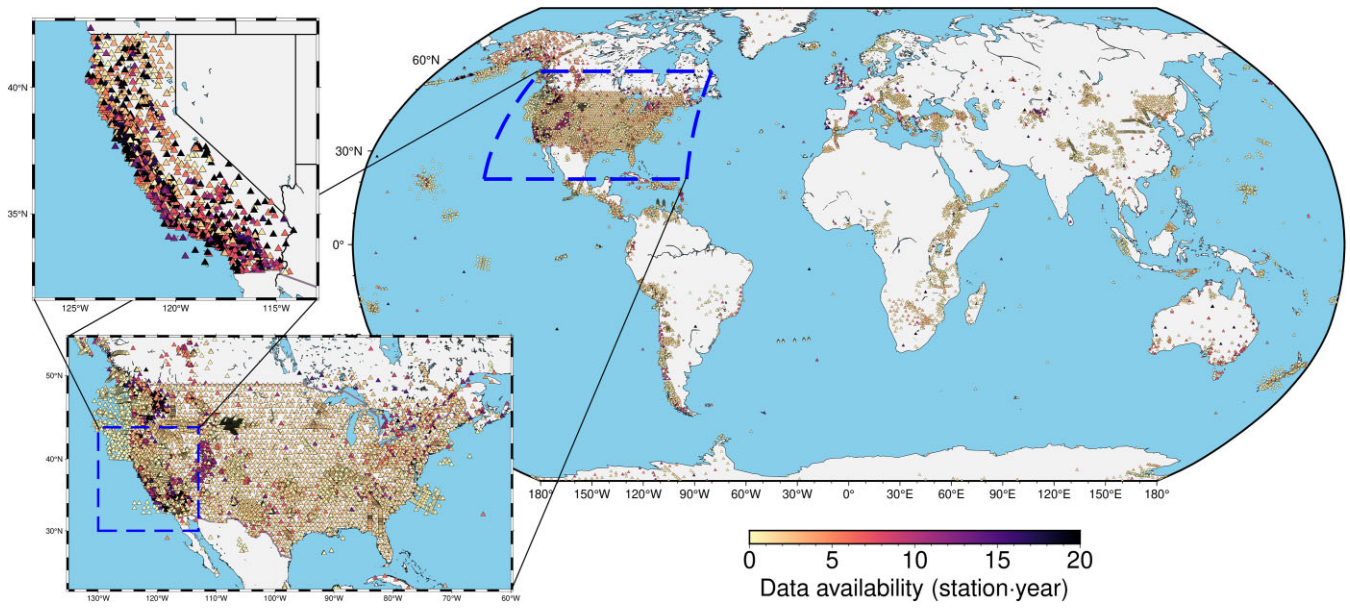

Figure 2. Station map of cloud-hosted data: A total of about 1.3 PBs of seismometre miniSEED data is hosted on Amazon Web Services (AWS) cloud storage: the EarthScope Consortium seismic data archive, and the Northern and Southern California Earthquake Data Centre. Each triangle indicates a seismic station and is colour-coded by its data availability.

moderate to large-sized data sets. However, on-premise storage and bandwidth limitations constrain point-to-point transfer efficiency. At the same time, the host takes responsibility for processing requests and maintaining stable data access, usually at the cost of additional man-power. Cloud storage provides an accessible solution for hosting and sharing scientific data with exceptional scalability and durability, along with improved findability, accessibility and reusability (Abernathey *et al.* 2021).

Cloud object storage offers a significant advantage by enabling the storage of large data sets and scalability for massively parallel queries from both within and outside the cloud. For instance, AWS provides the Simple Storage Service (S3), a scalable object storage service that allows users to store data as objects without practical limits on total data volume. Azure (Blob storage) and GCP also provide similar services. In contrast to the filesystem used in HPC, files are saved as individual objects and organized in a bucket with a flat structure. S3 objects can be identified by a prefix and a key, much like a folder and its filename in a filesystem, which users are more familiar with. While S3 is not a Portable Operating System Interface and differs from the Unix-style semantics most researchers are used to, the departure is advantageous: freeing storage from strict POSIX rules lets cloud object stores scale to billions of files, offer global access from any service and replicate transparently across regions.

Seismic data centres curate and offer vast amounts of invaluable data through the International Federation of Digital Seismograph Stations (FDSN) web services, which standardized API queries for seismic data and metadata delivery (Hutko *et al.* 2017; Hauksson *et al.* 2020; Quinteros *et al.* 2021b). Cloud storage fundamentally transforms data centre solutions, improving robustness, durability and stability to data management within the facility, resilience to storage read spikes and proximity between storage and compute nodes on the user end (Beckwith 2011). The SCEDC has been the pioneer in migrating, hosting and publicizing ∼150 TB of continuous data onto the cloud (Yu *et al.* 2021; Zhu *et al.* 2025), followed by the effort of NCEDC of their ∼190 TB, and most recently, the EarthScope Consortium seismic data archive, which has surpassed 1 PB (see Fig. 2). Direct access to cloud-based archives enables the development of cloud-native workflows, which we will discuss in Section 5. For instance, the workflow's throughput could not be achieved through the FDSN `fdsnws-dataselect` web service. However, we note that there are examples of petabyte-scale storage services with high throughput operated by scientific institutes, for example, for particle physics (Mkrtchyan *et al.* 2021).

Data centres migrating their repositories, or offering copies, in cloud environments must consider how data will be discovered and accessed directly by researchers. Whereas previously, these repositories were only accessible behind services that acted as abstraction layers, the organizations' data can now be exposed to direct access (e.g. back-end or raw files). This is important to avoid adding processing bottlenecks, limiting the ability to subset data and slowing down data queries and downloads. At the same time, such direct access increases the burden on the user to write efficient code, downloading only the relevant sections of data, a task previously handled by the access layer. Operational requirements like controlling access to restricted data, log data use, etc., may slow down the data access, and careful design must be considered depending on the application. New services and software are needed to support the efficient discovery and use of large-scale analyses beyond the simplest cases.

2.2 Data formats on object storage

2.2.1 Array data

Historically, data centres have relied on SEG-Y (for active seismology) and Standard for the Exchange of Earthquake Data (SEED, for passive seismology) as primary data formats (Guimarães *et al.* 2021). SEG-Y, designed initially for tape storage in the 1970s, remains widely used due to regulatory requirements in the petroleum

exploration industry. Still, it suffers from poor parallel read performance and high I/O latency in modern cloud environments. Meanwhile, SEED (or more accurately, miniSEED) is also actively used for seismic time-series archiving and shipping. Specifically, the miniSEED standard in its 2+ version adopted the paradigm that separates waveforms (i.e. miniSEED with minimal metadata) from their metadata counterpart (i.e. dataless SEED with no time-series). This only makes this format partially cloud optimized because it has no native support for object-based data partitioning, indexing or scalable metadata integration. It also requires additional infrastructure to serve efficiently from cloud storage (e.g. an indexing layer or a metadata catalogue).

More recently, the HDF5 format (Hierarchical Data Format version 5, The HDF Group 1997–2023) enables researchers to design their data structures and storage. HDF5 supports a customizable and flexible hierarchical schema, allowing the storage of multidimensional seismic waveforms and metadata (e.g. Krischer *et al.* 2016; White *et al.* 2023). On the other hand, such a self-describing structure also presents obvious limitations when used at scale in distributed cloud computing. The monolithic nature of HDF5 files introduces overhead for metadata handling and parallel access, while losing efficiency on byte-range requests and compressed blocks, that is, read subsets of large data sets (Ni *et al.* 2023). Despite efforts from open-source projects such as `h5coro` (H5 cloud-optimized read-only library, https://github.com/SlideRuleEarth/h5coro) and kerchunk (https://github.com/fsspec/kerchunk) that were made to optimize HDF5 for cloud object storage, they are often read-only solutions and do not fully resolve challenges from generalized file structures.

Recognizing these limitations, geospatial data initiatives such as Pangeo and EarthCube have pioneered the adoption of cloud-optimized array formats like Zarr and TileDB, which avoid hierarchical formats and instead store multidimensional arrays in a chunked, compressed and distributed manner. Pangeo's use of Zarr, for instance, has enabled massive parallel processing of gridded climate and remote sensing data sets, demonstrating ∼10x speedup in read performance compared to HDF5 when accessed in parallel from cloud object stores (Abernathey *et al.* 2021). Similarly, TileDB has proven effective for sparse geospatial data, including GNSS and seismic sensor arrays, allowing efficient subsetting and time-series access (Habermann *et al.* 2021). These technologies are now being adapted for seismology, where efforts such as those by Ni *et al.* (2023) demonstrate that converting DAS data from HDF5 to Zarr/TileDB results in significant memory and compute time improvements. The proliferation of these new open formats and their variants may be an obstacle to the sustainability of our software, so seismological workflows must stay format-agnostic and ready to pivot between, or simultaneously support, multiple storage layouts as standards evolve.

2.2.2 *Point sensors*

Seismological research spans a wide range of temporal and spatial scales, requiring storage solutions that can support both short-duration, multistation queries and long-duration, station-centric analyses. These workflows stress storage along different query axes: some pull data from thousands of stations but only short time windows (e.g. large-N arrays, ambient-noise cross-correlation), whereas others retrieve long, continuous histories—months to years—for each station in the network (e.g. ambient-noise monitoring, template matching). Small-object storage enables efficient writing and retrieval, often providing superior performance relevant to seismological broad-band seismic data (e.g. 10 MB per day for a typical 100 Hz 3-component broad-band data).

Cloud-optimized formats designed for petabyte-scale data sets—such as Zarr and TileDB—have emerged from the geospatial data community to support large queries on colocated compute, enabling efficient streaming of large data chunks. These formats are particularly advantageous for array-based data sets, including nodal deployments and DAS records (e.g. Ni *et al.* 2023). However, for many seismological applications, which typically involve small, targeted data requests, established formats like day-long miniSEED files remain highly effective. miniSEED supports efficient remote access to small waveform segments, making it well-suited for near-real-time, station-centric data streaming and event-driven workflows.

The data centres mentioned above all host seismic time-series in the miniSEED format on AWS S3. There are differences in conventions for organizing files in the S3 bucket regarding data granularity. Specifically, SCEDC and NCEDC store one channel per object, whereas the EarthScope seismic data archive groups all channels per station in a single file. Despite hosting more small objects, the former structure exhibits high efficiency when querying data from subset channels or locations since no redundant bytes are read. The latter structure usually requires an external data base that indexes files to facilitate data query (e.g. using mseedindex, https://github.com/EarthScope/mseedindex), although a fully cloud-native access library has yet to be developed. The various bucket structures indicate that object naming (prefix and key) has not yet reached a standard across these data centres because the new paradigm of direct access renders these previously hidden implementation details part of the user interface.

Delivering seismic station metadata in a cloud-native way remains an open challenge. At present, the FDSN web service for metadata still operates on on-premises servers at each data centre, creating a bottleneck that is not resilient to large-scale or burst queries. As an initial step toward addressing this, SCEDC and NCEDC have publicized community-standardized StationXML files—containing the complete history of station metadata and instrumental response—directly in their respective cloud storage. While this approach makes metadata accessible from the cloud, StationXML itself is not optimized for object storage: reading and parsing full XML documents is often slower and less efficient than targeted queries via the FDSN web service. Significant work remains to design and adopt metadata formats and delivery mechanisms that are truly cloud-efficient, enabling scalable, selective and low-latency access.

3 CLOUD DATA BASES

Seismologists need data bases to manage station metadata and curated data sets such as earthquake catalogues and phase picks. A data base is an organized data collection designed for efficient storage, retrieval and management. The simplest form of a data base, which is termed a flat-file data base, relies on files stored in hierarchical directories, where data is typically structured in plain text formats [e.g. Comma Separated Value (CSV) and JavaScript Object Notation (JSON)]. Although flat-file data bases are useful for small-scale or static data sets, their limitations in scalability, query flexibility and concurrent access become apparent when managing large volumes of data.

Modern data bases are broadly categorized into relational and NoSQL (non-relational Structured Query Language) data bases,

each addressing distinct needs. Relational data bases were developed decades before the advent of cloud computing. They organize data into structured tables of rows and columns governed by the relational model. The relational model handles structured data with a strict schema and transactional consistency, making it ideal for curated seismic meta data (e.g. event catalogues or station metadata) scenarios.

In contrast, NoSQL data bases became popular during the big data era to address the challenges of scalability, schema flexibility and heterogeneous data types, which are common in modern seismological research. NoSQL data bases include document stores (e.g. compatible with a tool such as MongoDB), key-value stores, wide-column stores and graph data bases. The MsPASS framework is an example of using MongoDB to manage large-scale seismic data (Wang *et al.* 2022). NoSQL data bases thrive in cloud environments due to their flexibility in managing semi-structured data (e.g. processed waveforms or phase picks) and scalability across distributed systems. This aligns seamlessly with the growing reliance of seismological research on high-volume, multimodal data sets, such as machine learning-ready archives.

Among NoSQL data bases, document stores like MongoDB and its AWS implementation, DocumentDB, exemplify the advantages of schema flexibility in modern seismological workflows. These systems store data as JSON-like documents, enabling researchers to consolidate heterogeneous data sets, such as workflow parameters, phase picks and semi-structured metadata, into a single data base without rigid schema constraints. This flexibility is particularly valuable in seismology, where evolving research workflows often generate metadata with new and inconsistent attributes. DocumentDB further simplifies scalability by automating sharding and replication in cloud environments, allowing distributed storage of large-scale data sets while ensuring low-latency access. Modern cloud data bases thus can store the massive output of seismic processing (e.g. billions of phase picks or cross-correlation measurements) and enable quick queries. We will explore the application of cloud-hosted data bases in seismology in the following cloud workflow examples.

4 CLOUD COMPUTE

Cloud providers offer diverse infrastructures and services well-suited to seismologists' diverse needs. Orchestrating these various cloud services to support seismological research involves integrating compute, storage and software tools provided by cloud vendors using service orchestrators such as CLI (command-line interface) tools, Python-based Software Development Kits (SDKs; e.g. `boto3`, `google-cloud-python`), or web-based platforms (e.g. AWS Step Functions, Google Cloud Workflows). Scientists developing workflows on the cloud face challenges when providers update their services and adapt open-source software, but this is possible with the help of research software engineers (Krauss *et al.* 2023). To detect and adapt to cloud service changes efficiently, ensuring workflow resilience requires version pinning, modular pipeline design and automated testing. On the other hand, the diversity of resources in the cloud, in particular, the variability of available machines and services, is typically much higher than in on-premise HPC systems. While numerous large-scale national science infrastructures exists,[1] they are typically much more homogeneous. For researchers, the cloud infrastructure allows them to tune the requested resources precisely to their demand. At the same time, this increased flexibility also comes with increased effort for identifying the optimal resource allocation.

The most basic unit on the cloud for scientists and many other higher level cloud services is a virtual machine (VM). VMs are the virtualization of hardware and packetization of operating systems, enabling users to access and share physical infrastructure on demand. The host operating system logically isolates different VMs running on the same machine. VMs can be configured to the user's preference, including the number of vCPUs, Random Access Memory (RAM), local storage and additional resources like GPUs, all selected from a set of provisioned templates. The flexibility of VMs is fuelling a democratization of large-scale computing. This review focuses on the type of parallelization well suited for cloud platforms, one of distributed memory, sometimes referred to as 'embarrassingly parallel.'

4.1 Batch computing

As a service commonly available in cloud systems, such as Azure, AWS and GCP, Batch computing involves the parallelization of jobs on cloud instances, similar to the job arrays in the SLURM scheduler system (Yoo *et al.* 2003). The batch computing service is particularly useful for parallelization when the job array shares the same code base and differs only in passing arguments, whether the jobs are simply command lines or containerized tasks. While the embarrassingly parallelizable job runs independently, the multinode parallel job allows internode communication through message passing libraries (e.g. MPI; Gropp *et al.* 1996). Such a framework enables single jobs spanning multiple computing instances as a cloud-based on-demand cluster for high-performance computation applications (Breuer *et al.* 2019; Zhuang *et al.* 2020; Dancheva *et al.* 2024).

Similar to the scheduler in a modern HPC system, an autoscaling mechanism dynamically provisions and scales cloud resources based on the volume and requirements of submitted workloads. Such a mechanism adjusts the number of running instances or containers to ensure that resources match the computational requirement without over-provisioning. Autoscaling can be triggered by pre-defined metrics such as CPU utilization, memory usage, request rates or job numbers, allowing cloud environments to handle traffic spikes and workload fluctuations efficiently. This capability is essential for maintaining high availability, improving fault tolerance and optimizing resource utilization and spending, making it a key feature in modern cloud infrastructure.

4.2 Serverless

Serverless computing is a cloud-native execution model that abstracts infrastructure management, allowing developers to deploy code without provisioning servers. By automatically scaling resources in response to demand, the serverless architecture enables users to focus on their applications rather than managing operational overhead. This is particularly transformative for seismic early warning systems, where latency-sensitive processing of real-time data (e.g. event detection) requires rapid, event-driven workflows. In a serverless framework, cloud providers like AWS (Lambda), Azure (Functions) and Google Cloud (Cloud Run Functions) dynamically allocate compute resources in response to user requests.

[1] e.g., Jean Zay (http://www.idris.fr/eng/jean-zay/jean-zay-presentation-eng.html) or JUWELS https://apps.fz-juelich.de/jsc/hps/juwels/index.html

A compelling example of serverless computing in seismology is demonstrated by Mohapatra *et al.* (2025), who tested a hybrid cloud-local workflow using AWS Lambda and the MsPASS framework. Their study found that downloading raw seismic data (i.e. the 40 million-record USArray data set) to local HPC clusters created untenable bottlenecks, requiring approximately 462 d for single-worker processing. By shifting pre-processing to serverless functions (e.g. noise reduction, metadata filtering), they minimized data transfer volumes and achieved throughput comparable to local HPC processing. The authors conclude that doing some or all processing on the cloud in this fashion will be essential for any processing involving large volumes of data already stored on the cloud. While hybrid workflows incur cloud costs, they bypass local network limitations, offering a scalable path for modern seismology.

4.3 Visualization

Effective visualization turns today's petabyte-scale seismic data into insight at a glance. Many groups rely on browser-based notebooks that connect directly to cloud object storage. Browser-based notebooks—commonly hosted in JupyterHub or Google Colab—can mount cloud object storage directly (e.g. using tools such as `s3fs` for AWS or `gcsfs` for GCP), read only the data chunk required for a plot and render interactive figures in real time. When those exploratory notebooks mature, researchers often containerize them into lightweight dashboards (such as `Dash` or `Streamlit`) that run on serverless platforms.

Because 'serverless' platforms start containers only at researchers' demand (e.g. during an earthquake crisis or a teaching lab), hosting costs remain a few USD per month in quiet periods. When visualizing full 3-D wavefields or large seismic data sets such as from DAS, teams spin up short-lived GPU instances running remote desktop tools such as `ParaViewWeb`, `PyVista` or cloud-proprietary software such as NICE DCV for AWS, to stream pixels, not raw data, to the user.

Once these visualization environments are running, the challenge becomes delivering data to them efficiently. Using cloud-optimized formats such as Zarr and TileDB, seismic data sets can be stored in small, independent chunks, allowing rapid retrieval of specific subsets—for example, waveforms from a given station and time window, or a horizontal slice through a 3-D wavefield—without downloading entire files. Low-resolution previews, map tiles or simplified 3-D model snapshots can be served instantly to support smooth navigation and exploration. More computationally demanding visual products, such as spectrograms, cross-sectional wavefield views or machine-learning-derived feature maps, can be generated asynchronously in the background, with only the resulting lightweight images or metadata returned to the user interface. This workflow enables seismologists to interactively explore data, compare results and monitor networks in real time through a web browser, while keeping bandwidth and latency low. The products of these visualizations still require robust data bases to store and organize derived measurements, as discussed in Section 3.

5 CLOUD-NATIVE APPLICATIONS IN SEISMOLOGY

Seismological analyses often require large computational bursts followed by long idle periods of analysing, a usage pattern that is tailor-made for on-demand cloud resources. Early adopters, therefore, re-hosted existing HPC pipelines in the public cloud to rent capacity only when peaks arose, a form of cloud-assisted 'lift-and-shift' workflow. Wang *et al.* (2018) was among the first to demonstrate the use of public cloud computing for seismic data processing at the TB scale and performed noise cross-correlation using the Aliyun cloud service, specifically with the Batch service and the cloud object storage. MacCarthy *et al.* (2020) used the FDSN web service to request data on the fly while detecting harmonic tonal noise. They successfully scanned 6 TB of USArray data within 4 d on a Kubernetes cluster with 50 EC2 nodes. Witte *et al.* (2020) built a serverless seismic imaging application ported from the HPC platform, dynamically scheduling jobs and provisioning computational resources using serverless cloud, demonstrating excellent cost efficiency, scalability and performance competitive with on-premise HPC clusters. Similarly, Zhu *et al.* (2023) designed an integrated earthquake detection workflow with containerized submodules, that is, data streaming, phase picking, association and event location. The autoscaling mechanism was implemented at both the cluster and cloud platform levels, enabling the automatic provisioning of computational resources based on job load. The paradigm of these cloud-based workflows was summarized by MacCarthy *et al.* (2020) as comprising three primary components: the infrastructure, cluster management software and domain-specific research software. These 'lift-and-shift' studies share a three-tier pattern: infrastructure (cloud VMs and containers), cluster management layer (Kubernetes, batch, step functions) and domain software (e.g. noise cross-correlation, full waveform inversion, earthquake detection). Computing efficiencies depends on how tightly these elements are coupled.

However, these workflows assumed that data could be fetched on demand, but pulled data from outside the cloud; such a pre-requisite cannot be easily met in the cloud for data-intensive tasks. Transferring the raw data or data products can be time-consuming and expensive (Wang *et al.* 2018; Ni *et al.* 2023). For example, Zhu *et al.* (2023) spent ∼50 per cent of total job time downloading waveforms through the FDSN web service. Requesting data at scale may also pose challenges for data centres that receive unpredictable heavy traffic and clusters where big data is saved and managed on-site. To overcome these bottlenecks, recent efforts embrace 'cloud-native' workflows: (1) harness direct access to the cloud-hosted data to avoid copying data (MacCarthy *et al.* 2019; Yu *et al.* 2021), (2) leverage cloud-managed services and (3) employ containerized software to ensure portability, consistency and ease of deployment across platforms. To illustrate the cloud-native workflows for seismology, we present two contrasting research workflows in seismology that have benefited from the cloud systems: (1) cross-correlation for ambient noise seismology and (2) earthquake catalogue building workflows. Fig. 3 illustrates the two alternative workflows with data flows and associated cloud services best used in large-scale jobs.

5.1 Workflow 1: Large ambient noise seismology

Ambient noise seismology is the methodology that utilizes continuous seismograms, typically dominated by a diffuse, ambient seismic field, to extract spatial or temporal variations in seismic wave speeds. This method has been widely used for (1) Earth Imaging reconstruct high-frequency Rayleigh wave and image shear wave structure where seismic stations are located (e.g. Shapiro *et al.* 2005) and no longer rely on rare earthquakes, and for (2) Earth monitoring by exploring changes in subsurface structure by subtle

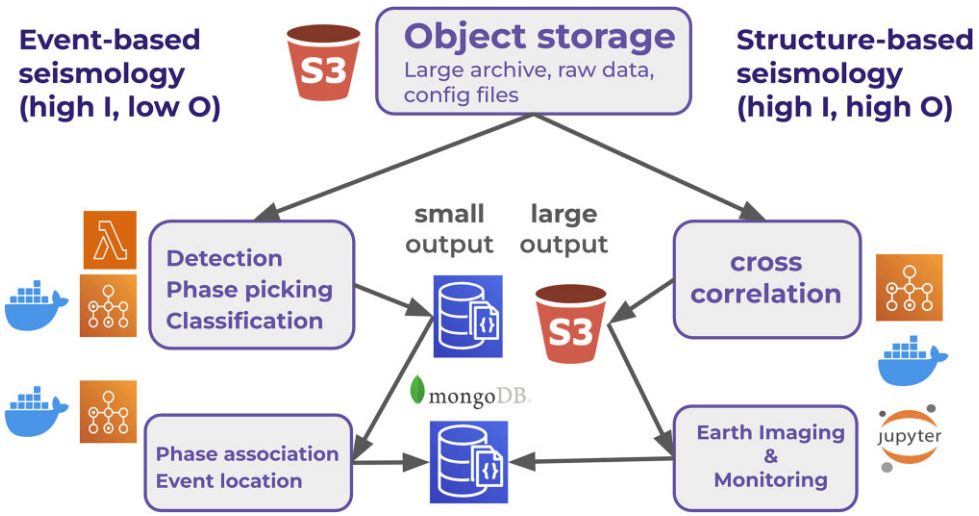

Figure 3. Two canonical workflows in seismology: Event-based seismology that reads large volumes of seismic data (high I–high Input) but outputs low volumes of data in data bases (low O–low Output), and structure-based seismology that reads large volumes of seismic data (high I) and outputs large volumes of seismic data (high O). The first workflow outlines the basic steps in generating a seismic event catalogue. The second workflow describes how to extract seismic properties of the subsurface with ambient field seismology, which generates high data volumes of ambient noise cross-correlations.

phase shifts on the coda of cross-correlations (e.g. Sens-Schönfelder & Wegler 2011).

The method relies on the cross-correlation of short time-series between channels, presenting some of the most significant challenges in computational seismology. The cross-correlation typically uses short windows, ranging from minutes to a few hours, recorded at pairs of seismic channels and stacks these over days to years of data. Thus, the workflow scales quadratically with the number of channels N, a step that favours shared memory processing, and linearly with the number of windows to stack T, a step that favours distributed memory processing. The rise of array seismology with $N > 100$ is a real computational challenge. Because the workflow often entails storing cross-correlation functions, including intermediate steps such as substacking, cross-correlation may involve writing TBs of files. Several efforts have been made for open-source and large-scale computing of ambient noise cross-correlations, some leveraging CPU-based clusters (e.g. Jiang & Denolle 2020; Makus & Sens-Schönfelder 2024), others leveraging heterogeneous computing with CPU and GPU (e.g. Fichtner *et al.* 2017; Ventosa *et al.* 2019; Clements & Denolle 2020; Zhou *et al.* 2021).

Given two canonical seismological approaches for *Earth imaging* and *Earth monitoring*, the workflow to compute cross-correlation functions is multistep, and their optimal parallelism strategies differ. First, the cross-correlations are performed independently on synchronous time-series, which permits distributed memory parallelism, often referred to as 'embarrassingly parallel,' and scales only with the overall period of the instrumental record. Secondly, the cross-correlations are done on *pairs* of seismic channels, and a given channel window of data could be read once and cross-correlated over all other channels with $N(N-1)/2$ pairs. This strategy often employs multithreading with shared memory, parallelization across channel pairs, and leveraging GPUs to accelerate the correlation step (Fichtner *et al.* 2017; Clements & Denolle 2020). When the data is too large for the memory available, local storage of intermediate products, such as the Fourier transforms (Wang *et al.* 2018), or low-rank factorizations (Martin 2019), and parallelization over groups of station pairs is also possible (e.g. C4 project Schmitt *et al.* 2020, 2025).

Cloud infrastructure is particularly well-suited for ambient noise seismology, given its significant data throughput (reading and writing) and parallelization capabilities. Several attempts to perform ambient noise cross-correlations on the cloud demonstrated the speed and scalability of adapting cloud infrastructure. Wang *et al.* (2018) developed a parallelization scheme to independently calculate groups of channel pairs and perform massive daily cross-correlations, totaling 300 M, over 10 hr of processing on nearly 1000 virtual machines using the Aliyun cloud service. Ni *et al.* (2023) performed DAS cross-correlations on AWS using cloud-native workflows and achieved 300 M of daily cross-correlations over 64 instances in 24 hr, spending less than 20 USD. Clements *et al.* (2020) and Schmitt *et al.* (2020) developed a workflow on AWS that approached 'cloud-native' by streaming from S3 to EC2, generating cross-correlation locally, saving the results on disc and uploading them to S3.

We now present **Cloud-Native NoisePy**, a new version of Jiang & Denolle (2020) that has been updated with I/O for cloud-based data archives, enhanced object-oriented Python programming, including parallelization, flexibility in computing platforms and continuous integration. NoisePy leverages cloud object storage for massive I/O parallelization and short-term storage of temporary data, such as daily cross-correlation functions, as illustrated on the right-hand side of Fig. 3. NoisePy employs two primary parallelization strategies to optimize performance. The first approach utilizes the Batch compute service to execute each daily job of processing and inter-channel cross-correlation concurrently and independently, using the same container with different data. Within each job, NoisePy utilizes native Python multithreading for parallelization across several steps, including reading data, pre-processing, computing the Fourier transform, cross-correlation and writing daily results, which are stacked and saved as compressed NumPy `.npz` files back to storage (file structure and format were found optimal when experimenting with AWS S3). After processing all daily data, a final aggregation step combines the results to produce long-term correlation stacks. This architecture efficiently handles large-scale data, generating TBs of cross-correlation outputs in the cloud.

We present the results of an experiment in which we ran NoisePy on one year of SCEDC data from 2022 January 1 to 2023 January 1.

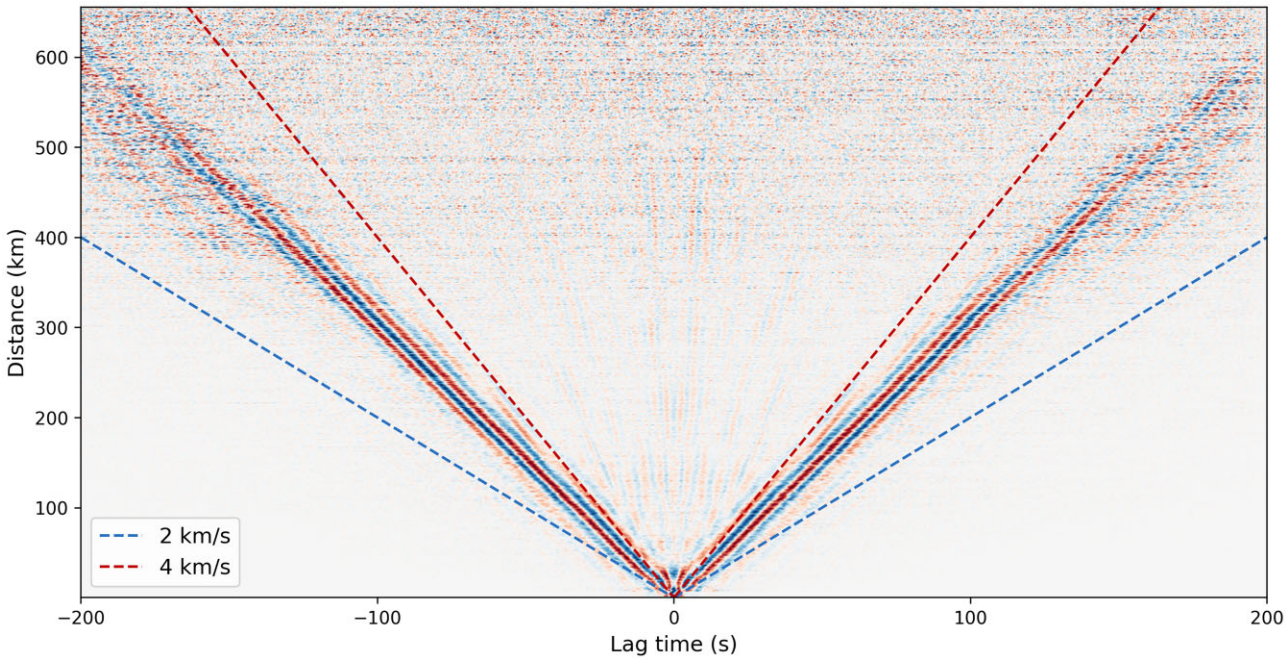

Figure 4. Ambient noise cross correlation using 1 yr of data from the Southern California Earthquake Data Centre: all data are publicly available in the SCEDC AWS cloud storage (s3://scedc-pds). The Z-Z component cross-correlation functions are bandpass filtered between 1 and 10 s.

We cross-correlated all of HH? channels that included 288 stations and about 43 000 station pairs. Ran on AWS Fargate with up to 64 instances, it took 11 hr of compute time to generate 6.2 million files of daily-stacked cross-correlation, with a volume of 1.6 TB on S3, spending about 250 USD on SPOT pricing. The second step on a similar Fargate cluster took 1.5 hr and generated the final stacks of all interchannel cross-correlations, totaling 23 GB of data and 46 000 S3 objects. We present the results of the 1-yr stacked ZZ component of the cross-correlation, bandpass filtered between 1 and 10 s, sorted by the interstation distance in Fig. 4. We find the convergence of the correlation functions past 600 km of interstation distances. Seismic tomography from these data products will involve extracting phase and group velocity measurements from these cross-correlations and inverting the frequency-dependent velocity curves into a shear-wave velocity model. Our experiment demonstrates the case of scalable data processing for tomography applications.

5.2 Workflow 2: Earthquake catalogue building

Earthquake catalogue building is a complicated, multistep workflow that ingests raw time-series data and outputs point clouds of earthquake locations and their attributes. The main steps are to detect events, identify the time at which seismic phases (typically *pickingP* and *S* arrivals) arrive, associate them with a specific origin (event), possibly incorporate location using 3-D Earth velocity models, relocate them using double-difference relocation and calculate source parameters, such as magnitude and focal mechanisms. Each step has been explored using machine learning. In particular, for phase picking, deep learning has proven highly successful, with models such as U-Net (e.g. Zhu & Beroza 2019; Münchmeyer *et al.* 2022). Network-based analysis typically requires gathering multiple-station data simultaneously, and has benefited greatly from U-Net and graph networks (e.g. Münchmeyer *et al.* 2021; Sun *et al.* 2023; Clements *et al.* 2024). Most workflows are a sequence of modules (e.g. Walter *et al.* 2020; Retailleau *et al.* 2022; Zhang *et al.* 2022; Zhu *et al.* 2023), where modules can be adapted according to the user preferences.

The computational efficiency of these workflows matters when considering large-scale deployment. Similar to the first workflow presented, we break down the computational efforts into two types of parallelization. The first step involves extracting features from raw data, including the arrival times of *P* and *S* waves, the amplitude and possibly the polarity of these waves. This process can be independently calculated for each data window as a separate job and is amenable to massive parallelization. The second processing step requires aggregating these features across the stations and benefits from multithreaded parallelization. Both steps can be easily orchestrated on cloud systems, which was first pioneered by Zhu *et al.* (2023) by including deep learning phase picking, association and relocation using double difference, and by Pierleoni *et al.* (2023) for Internet-of-Things (IoT) early warning systems whereby picking is done at the seismometer level, and location is done on the cloud. We present here another cloud-native workflow for the basic steps of earthquake catalogue building, utilizing the `SeisBench` ecosystem (Woollam *et al.* 2022) for phase picking, which we refer to as QuakeScope.

QuakeScope orchestrates its seismic catalogue workflow entirely on AWS Batch (Fargate) and a MongoDB-compatible DocumentDB cluster for outputting detection attributes and checkpointing. The basic unit of the workflow is a Python job, composed of four steps: (i) obtain a day-long time-series of miniSEED data, (ii) process the continuous waveform with a phase picking model implemented in SeisBench, such as PhaseNet (Zhu & Beroza 2019) or EqTransformer (Mousavi *et al.* 2020), (iii) remove the instrumental response to extract the amplitude of each detection, (iv) write the resulting detections to a DocumentDB. As jobs (i) and (iv) are I/O-bound, while jobs (ii) and (iii) are compute-bound, we implemented an asynchronous processing using the asyncio Python module. This means that the job processes data simultaneously, loads the next day and still writes the picks from the previous day. This increases throughput substantially and thereby reduces resource costs. The

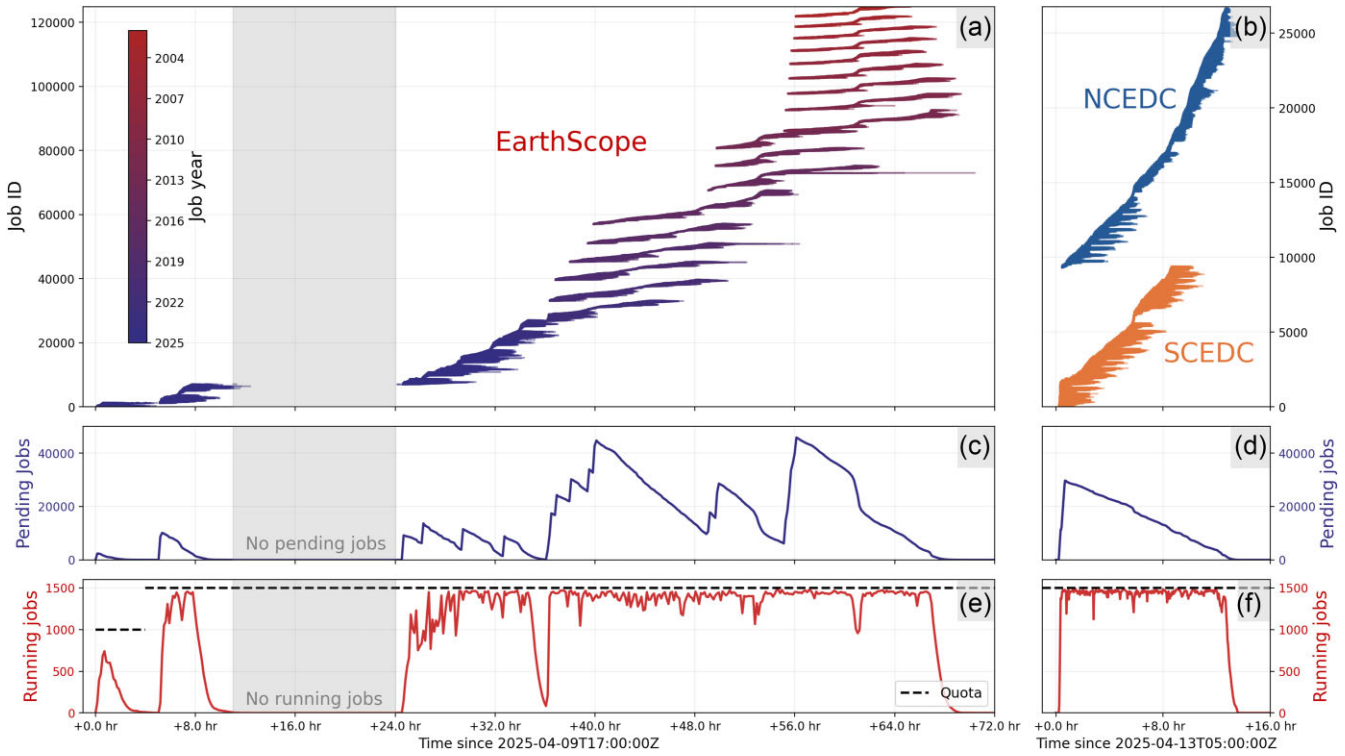

Figure 5. Evolution of job statistics in the 2025 Data Mine experiment: top panel has a line between the start and end time of each job that ran on Fargate. Each year's worth of data is sent manually as a separate batch of jobs, colour-coded for EarthScope. The middle panel shows the number of pending jobs in the Fargate queue. Jobs enter the queue and are scheduled to run until the quota of 1500 jobs is reached. The dips in the number of active jobs are attributed to our slower job orchestration. One that may be attributed to SPOT's intermitency is at hour 60. Upon success, the experiment on the NCEDC and SCEDC went smoothly, maximizing the number of jobs until completion.

individual steps communicate through limited-size First-In-First-Out queues to avoid excessive memory overheads.

We present the results of the 2025 Data Mine experiment, where we scanned the complete EarthScope Consortium seismic data archive (1 PB) and the SCEDC and NCEDC AWS-hosted open data, each with 150 TBs, with QuakeScope. We describe in Fig. 5 the evolution of our jobs on AWS. Our AWS allocation had limited quotas for 1500 jobs (12 000 vCPUs) to be used simultaneously, a tripled quota relative to the default limits. We manually launch each year, from 2025 going back in time, as a queue of jobs, and experimented with the EarthScope back-end servers as a pilot experiment. During the initial stage, we stress-tested the EarthScope archive. We progressively launched the jobs on Fargate SPOT (a lower cost queue that is less 'on-demand'). In the EarthScope experiment, our progressive load of jobs in the queue demonstrated the resilience of the backend system, allowing us to launch a larger batch of jobs by hour 40 in the experiment. For the NCEDC and SCEDC experiments with a smaller data set of approximately 300 TB, we launched all jobs simultaneously, reached our quotas, and completed them in 12 hr. The performance of cloud systems remained constant, demonstrating that this process can be easily accelerated by allowing for more concurrent processes. See Ni *et al.* (2025) for a complete report of this experiment and the data product.

6 COMMUNITY MODEL VISUALIZATION AND DISSEMINATION

Within the seismological community, there is an acute need for models of natural systems to support both basic and applied research. For example, models of seismic wave velocities within the Earth are essential for interpreting tectonic history and evolution (Delph *et al.* 2021), for generating accurate ground motion estimates using earthquake simulations (e.g. Glehman *et al.* 2024), and for enabling a wide range of analyses related to seismic hazard and risk. Similarly, geometric models of earthquake-producing fault structures, along with associated metadata such as slip rates and earthquake histories (Plesch *et al.* 2024; Seebeck *et al.* 2024), form the foundation of seismic hazard analysis (Petersen *et al.* 2020). These needs have driven the development of community models, which are collaboratively developed, synthesis-based representations of natural systems that are maintained and shared by a broad group of researchers with expertise in the relevant domain (e.g. Aagaard *et al.* 2025). Community models are typically open-source and publicly available, designed to incorporate the best available data and understanding and intended to serve as common foundations for scientific research, education and practical applications (e.g. Shaw *et al.* 2015; Plesch *et al.* 2024). The generation of community models is a focus of various community science organizations such as the Statewide California Earthquake Center (SCEC) and the Cascadia Region Earthquake Science Center (CRESCENT) (Melgar *et al.* 2024; Aagaard *et al.* 2025).

To enable FAIR (Findable, Accessible, Interoperable and Reusable) access to community models (Wilkinson *et al.* 2016) and facilitate their use across a wide range of applications, these models are often distributed in multiple formats—such as structured text files, binary volumes, data bases or web-based APIs—to accommodate diverse user needs and computing environments. However, the complexity and size of these data sets—particularly 3-D and 4-D models of Earth's interior or fault

systems—can present significant technical barriers for many users (Small *et al.* 2017). There is a growing need for software tools that allow users to interactively visualize, query and subset models before downloading or integrating them into workflows. These tools must balance performance with accessibility and support interoperability with widely used scientific programming languages (e.g. Python) and data standards. Ultimately, the effective use of community models depends not only on their scientific rigour but also on the availability of user-friendly software that lowers the barrier to entry.

The cloud is particularly well suited for the hosting, visualizing and disseminating community models because it provides centralized, flexible infrastructure that meets the needs of distributed scientific teams (Gentemann *et al.* 2021). Community models are often large, dynamic data sets and are accessed by users across multiple institutions. These characteristics make local storage solutions inefficient or inaccessible. Cloud storage enables elastic scaling, allowing both storage and computing power to dynamically adapt to the community model's growth, without the managerial burdens associated with on-premises infrastructure. It also facilitates consistent versioning, access control and metadata management, which are required for transparency and reproducibility. In addition, cloud platforms integrate seamlessly with modern computational tools and workflows, enabling users to analyse and visualize models directly in the cloud without transferring large data sets. As such, the cloud is a natural fit for managing community-driven, data-intensive geoscientific resources.

As an example of such a tool in the seismological community, the CRESCENT Community Velocity Model (CVM) Viewer and Repository (Bahavar *et al.* 2025a) is a cloud-based platform for storing, distributing, analysing and visualizing seismic wave velocity models of the Earth. It combines Python-based tools with a geospatial web interface to enable real-time, interactive exploration of data sets. By adhering to widely accepted metadata and file format standards, the platform ensures that data sets remain consistent, interoperable and ready for use in both research and education.

The CVM Viewer is a web-based geospatial visualization tool built with Python, FastAPI and CesiumJS (Consortium *et al.* 2018). It allows users to explore CVM data sets interactively through a cloud-hosted 3-D map. In addition to visualizing the spatial extent of seismic velocity models, the viewer displays known faults and earthquake hypocentres. Users can toggle terrain layers, adjust model boundaries and navigate using rotation, zoom and pan controls. Visualization tools include horizontal slices, vertical cross-sections and depth profiles, offering intuitive ways to investigate subsurface structures.

The CVM Repository hosts multiple user-submitted seismic velocity models that have undergone peer review and are published. Models are stored in netCDF-4 Classic and HDF5 formats on AWS S3, organized hierarchically to separate 3-D model volumes and associated surface data. Automated compliance checks are performed before storage to ensure alignment with community metadata standards. The backend, built with `xarray` and `h5py`, handles data queries and retrieves model subsets based on user-defined geographic and depth ranges.

Users can extract horizontal slices, cross-sections or full 3-D volumes in various supported formats. These extraction and conversion tools are deployed using AWS Lambda, enabling efficient access to large netCDF or HDF5 files and seamless format conversion. Throughout these operations, geospatial metadata is preserved to ensure compatibility and compliance with metadata standards. The entire system is deployed using AWS Fargate, a serverless container platform that automatically scales computing resources in response to user demand.

In addition to the CVM Viewer, CRESCENT is developing a suite of complementary cloud-based tools to support the broader earthquake science community. The CRESCENT Community Fault Model Viewer (Bahavar *et al.* 2025b) enables interactive visualization and dissemination of fault geometries and associated metadata. Other tools currently under development include platforms for storing, analysing and distributing paleoseismic data and seismicity catalogues, all designed with an emphasis on scalability, accessibility and adherence to community data standards. These emerging tools demonstrate how cloud infrastructure makes data and computational resources more accessible and promotes collaboration across disciplines and institutions.

7 EDUCATION AND TRAINING INITIATIVES

Open-source software and interactive cloud-based computing environments have transformed seismological research and education by providing free, accessible data analysis and modelling tools. Platforms like Binder, Google Colab and institutional JupyterHub instances enable researchers and students to run open-source Jupyter Notebooks without requiring local installation, significantly lowering entry barriers. `Seismo-live` (Krischer *et al.* 2018) is one example, offering a library of seismology-focused Jupyter Notebooks that can be executed directly in a web browser using Binder. Similarly, Google Colab provides a free cloud-based notebook environment with pre-installed libraries, enabling students to analyse seismic data sets and run numerical simulations from any device. Google Colab is often linked in seismological software repositories to provide tutorials on free cloud services (e.g. in SeisBench; Woollam *et al.* 2022). By leveraging these free Jupyter-based platforms, the seismology community ensures that computational tools are widely available, fostering open science, reproducibility and equitable access to high-performance research workflows.

Research projects may also lead to developing cloud-based workflows, and researchers may choose to provide guides for cloud usage. For example, Krauss *et al.* (2023) utilized the Azure platform and compared pre-trained Machine Learning models and template matching for constructing an earthquake catalogue offshore. Their work also provided informative, educational guidance for individual researchers in code development and containerization, cloud infrastructure, job design and performance analysis.

In higher education, instructors leverage cloud platforms to create interactive, scalable learning environments for seismology and data science. Universities may deploy cloud-like infrastructure with virtual machines for classroom instruction, enabling students to access the course with affordable devices (e.g. tablets, laptops or even phones) and run code and homework on cloud instances. Such cloud integrations enhance accessibility, enabling students to work with real-world data and modelling problems in an educational setting.

Hands-on training workshops are essential for advancing computational seismology skills, as they immerse participants in using modern software and large-scale computing resources. Recent community initiatives, such as the NSF-funded SCOPED project, have organized multimodal workshops to teach researchers and students how to use research-grade seismological software on cloud platforms (Denolle *et al.* 2025). Engaging the community in multiple ways with cloud infrastructure has been beneficial: from running simple workflows on a provided cloud-based JupyterHub

(e.g. the GeoLab workshop) to deploying an EC2 instance on their own (e.g. in HPS). Recent workshops have been dedicated to training several hundred participants, primarily graduate students, postdoctoral researchers and research scientists.

8 DISCUSSION AND OUTLOOK

8.1 Software as a service

Cloud infrastructure lets seismologists rent powerful computing services only when needed, but those virtual servers still start life as 'bare-bones' operating systems. Researchers must rebuild their software stack—libraries, compilers, scripts–every time they launch a new instance. To simplify this process, the common practice is to replicate working environments facilitated by environment management software, such as a lightweight solution of the package manager Anaconda (https://anaconda.org), or utilize a fully self-contained option such as Docker (Merkel *et al.* 2014) and Singularity (Kurtzer *et al.* 2017). Cloud platforms also provide ready-to-use virtual images that cater to general or geophysics-specific needs, often at a small additional cost.

A more user-friendly approach is Software-as-a-Service (SaaS). It is a delivery model that enables users to run scientific software easily while interacting through a web form or API. The back-end software can be cloud-optimized, and computing resources are provisioned upon users' request. Such a model enables users to access the software in a serverless setting without tedious configuration, while being elastic and cost-efficient for service providers. For example, Chen *et al.* (2013) proposed a web application that allows users to submit requests to generate synthetic seismograms. The service receives requests along with source parameter settings and initiates the 3-D elastic wave equation solver on the back-end. Researchers of interest may utilize this service through a direct and convenient web interface, receiving synthetic seismograms without requiring any software configuration. The trade-off is flexibility—custom methods or novel algorithms still require direct access to code and data, which SaaS platforms may not expose.

8.2 The self-imposed open and reproducible science

Seismology already enjoys a culture of open data and open-source software—community archives expose waveforms through standard FDSN web services, and libraries such as ObsPy (Beyreuther *et al.* 2010) and SeisBench (Woollam *et al.* 2022) make analysis scripts widely shareable. Yet, the full research replication requires running the full-stack workflow on any machine to obtain similar research results.

Cloud computing has become a critical enabler of reproducible research by forcing workflows to be explicit and portable. In traditional observational seismology, workflows often rely on trial and error, such as manually selecting bandpass filters based on domain expertise and visual data inspection. While this approach benefits from expert judgment, it poses significant challenges to reproducibility—other researchers may struggle to replicate results if they do not use the same parameters or follow the same steps. Cloud computing, however, requires the creation of well-defined, version-controlled and containerized workflows that are portable and executable in consistent environments. This shift towards standardized research workflows facilitates reproducibility by ensuring that the exact conditions under which the research was conducted can be easily replicated on different systems or by different researchers. This self-imposed reproducibility fosters a more rigorous and transparent scientific process, crucial in modern seismology, where big-data applications increasingly dominate.

8.3 The cost of the cloud

Most cloud service providers employ a pay-as-you-go pricing model, where users are only charged for using any related cloud resource. Because commercial cloud services are not directly sponsored or subsidized by government funds, their cost is priced by hardware rental fees, power and facility fees, and commercial profit, but are comparable to the 'fully burdened' HPC costs after removing government subsidies. Researchers can estimate the cost of their workload to the first order by timing the duration of time that virtual machines run and the time that data is stored in cloud storage. For example, we summarize the spending of Clements & Denolle (2023) based on AWS EC2 and S3 pricing policy: (1) downloading and uploading 50 TB of NCEDC data (∼40 USD), (2) performing the single-station noise correlations (<50 USD) and (3) storing all data over one week (∼40 USD per week). Such pricing model holds cost advantages: (1) cloud resources are accessible to almost everyone, whilst on-premise equipment makes one-time spending unnecessary, (2) maintenance is performed by cloud service providers instead of full-time institutional IT employees and (3) spending is better quantified and monitored through the billing statistics and may help future budgeting (Norman *et al.* 2021).

However, the pricing model can be complex when chargeable usages are vaguely defined for some services and resources. Consequently, budgeting for cloud infrastructure beyond the basics is often more complicated. For example, in the cloud-native NoisePy test (see Section 5.1), we utilize AWS S3 to save pre-stacked correlation functions before stacking. Besides the ephemeral storage cost from pre-stacked correlation functions, S3 write (PUT) and read (GET) operations also come at a flat rate (usually several USDs per thousand requests). Despite being minimal, such cost shall not be omitted when performing a large-N and large-T ambient noise interferometry study with potentially millions of such operations. Moreover, data bytes travelling across cloud regions will incur a perceivable egress cost, unless waived under specific agreements [e.g. the Open Data Sponsorship Program, which supports the SCEDC S3 archive presented by Yu *et al.* (2021)]. Additional charges may also be applicable for on-demand servers. For instance, I/O operations with the standard pricing model are not free for DocumentDB clusters. With this type of cluster, spending may be significant when I/O usage is extensive (e.g. phase picks, insertions and metadata queries that scale with the job, see Section 5.2). Such scaling terms should be identified and optimized to avoid unexpected spending in a cloud-native workflow. Despite these challenges, our experience has been fortunate to overbudget, strategize to optimize and conclude with a lower overall cost.

Explicitly comparing cloud compute costs to the costs for on-premise HPC is challenging, as numerous factors need to be taken into account for on-premise HPC, such as procurement cost, energy, facilities and labour. In addition, such costs will vary strongly on a country-by-country basis. However, for a rough estimate, we compare the costs for AWS Fargate service, the service used for the phase picking workflow above (Section 5.2), to estimates from different universities. A CPU core hour (SPOT instance as in our workflow) on AWS costs around 0.012 USD (∼0.010 EUR). The

costs charged (for internal users) by different universities vary between 0.008 EUR and 0.012 EUR.[2] We note that the charge for main memory on the university deployments was overall slightly higher than on AWS Fargate; however, the cost is dominated by the cost per CPU core. While large-scale users should consider a more precise, task-specific assessment of costs, taking into account local conditions and the expected long-term resource utilization, these numbers illustrate that for workflows as the ones presented, the cost of on-premise and cloud resources is in a similar range. In all instances of researchers using institutional computing centres or cloud services, federal research grants can support these costs [e.g. as was the case for the numerical experiment shown here through a National Science Foundation CloudBank grant (Norman *et al.* 2021)].

In summary, researchers in seismology will benefit cloud resources if (1) they need on-demand resources as in early warning or rapid post-event response without provisioning them over long time-frames, (2) many researchers are working on the same data (e.g. shared open-access data archive or post-earthquake response archive) and (3) the data archival is supported by cloud providers (e.g. AWS open data), non-profits (e.g. RadiantEarth https://radiant.earth/) or funded by governments [e.g. National Science Foundation Seismological Facility for the Advancement of Geoscience (SAGE) facility, now operated by the EarthScope Consortium] and on the order of peta-scale. Cloud services may be too costly for researchers who work directly with TBs of data, whose analyses are not time-constrained, who can guarantee a stable utilization of on-premise resources and who mostly want to share that data within their institutional research groups (i.e. a limited number of users).

8.4 Disadvantages of cloud computing for science

No medium offers true permanence—disc drives typically require replacement after 5–8 yr—and cloud object stores are no exception. The difference is economic rather than technical: keeping a petabyte of data in a frequent-access state becomes an annual operating expense that can outstrip the one-off capital cost of a local tape library, especially once data-egress and retrieval fees are factored in. Although a common perception is that commercial clouds are less reliable than institutional computing centres, the underlying hardware is identical. The fundamental trade-off, therefore, is not reliability but the ongoing cost model and the penalties for frequent, bulk access to archival data sets.

Moreover, because cloud uses shared but logically isolated hardware, run-to-run performance may fluctuate as neighbouring users compete for network or disc bandwidth, whereas a dedicated on-prem workstation delivers near-identical wall times each day. Secondly, tightly coupled jobs still run more efficiently on institutional clusters wired with low-latency node communication; comparable services do exist on the cloud, but the hourly premium usually outweighs any elasticity benefit. Finally, the funding model feels unfamiliar: instead of drawing on a pre-paid allocation of CPU-hours, researchers receive a transparent bill that details costs, which can be unsettling when project budgets are built around fixed grant lines.

[2]https://hpc.rtu.lv/hpc/hpc-services/price-list/?lang=en, https://hpc.ut.ee/pricing/calculate-costs, https://hpc-docs.uni.lu/policies/usage-charging/ (last access 2025 August 11 for all)

9 CONCLUSION

Cloud computing and storage offer a viable solution for analysing petabyte-scale seismic archives, leveraging the concept that data is transferred only once but distributed to many users. Cloud-optimized and containerized workflows can be spun up on demand at a relatively affordable cost for the size of the task. Cloud computing and storage solve the problems of slow campus Ethernet networks and shared HPC queues, enabling peta-scale analysis completed in hours to days, instead of months. However, building a cloud-ready pipeline remains a significant undertaking for most scientists. Containerizing legacy code, wiring up object storage I/O and automating provenance all require skills that fall between classical research and production software. Research software engineers remain essential: they translate scientific intent into robust, version-controlled artefacts, add automated tests and keep pace with the rapid evolution of cloud services. Once that up-front investment is made, every run is self-documenting and trivially repeatable, lowering the long-term maintenance burden. Prototyping for cloud systems on local servers is beneficial (e.g. minimizing cost) and possible (e.g. developing reproducible software and environments), for example, using tools such as MinIO object storage systems and MongoDB data bases. Researchers may focus on workflow reproducibility, open-sourcing software and minimizing job requirements to increase speed and lower computing and environmental costs.

Our performance benchmarks illustrate the payoff. In our ambient-noise test case, 1.6 TB of correlation functions—6.2 million files—were generated in eleven hours for about 250 USD, all without touching a single on-premise disc. Likewise, the QuakeScope catalogue builder scanned roughly 1.3 PB of global data sets in three days, limited only by computing quotas, and likely discovered 10 times more earthquakes than previously reported. Automatic retries on pre-emptible instances kept utilization high and human intervention low, demonstrating that today's managed services can rival dedicated HPC for embarrassingly parallel seismology at comparable cost.

Looking ahead, both DAS and the rapid proliferation of low-cost IoT seismometers will significantly increase data volumes and latency requirements, far exceeding today's norms. In the earthquake early warning architecture, hundreds to thousands of edge devices can run lightweight pickers locally, then stream only compact, parametric data to the cloud. There, serverless functions fan-in those messages, trigger association and localization and broadcast alerts—often within a few seconds of the origin time. The rapid scaling of cloud resources is particularly suitable for handling rare and extreme events. Finally, Cloud providers' per-region carbon-intensity dashboards (e.g. AWS's Sustainability Pillar, Google's Carbon-Free-Energy scores, Microsoft's 2025 renewables target) can inform users to choose cloud computing regions with a lower carbon footprint without sacrificing latency and performance.

However, there are caveats about using the cloud for seismological research. Cloud services are expensive for direct prototyping and experimenting with orchestrating autoscaling. Given the investment in software development, it is also not necessarily advantageous to use the cloud for analysing data sets on the order of 100 TBs or less. Furthermore, the cost of cloud storage is significant for individual researchers, except when benefiting from free open data programs (e.g. AWS open data), and when national facilities that typically invest heavily in on-premise servers find cost benefits in migrating services to cloud systems (e.g. EarthScope). Finally, technological advancements in commercial clouds are needed to compete with the tightly coupled nodes of conventional HPC centres.

Realizing that vision will take community effort—shared, containerized code, FAIR data in cloud-optimised formats and collaborations between domain scientists and research software engineers. Computing skills and software best practices are constantly improving, especially with the rise of open-source programming languages and training materials (e.g. Community 2025).

Publicly funded research networks such as ESnet (United States), GÉANT (Europe) and CSTNET (China), paired with high-speed transfer tools like Globus (Allen *et al.* 2012) and GridFTP, enable researchers to move multi-terabyte to petabyte data sets at tens to hundreds of gigabits per second, with record demonstrations reaching 1 PB d^{-1}. These dedicated optical backbones interconnect supercomputing centres, data archives and universities, providing low-latency, high-reliability transfers at no direct cost to scientists, critical for sharing seismic, satellite and climate model data. In parallel, the compute-close-to-data paradigm allows researchers to run tens of thousands of jobs directly adjacent to petascale data sets on public HPC platforms [e.g. United States Department of Energy facilities, Partnership for Advanced Computing in Europe (PRACE), China National Grid (CNGrid)] or commercial clouds, minimizing transfer time entirely. Public HPC resources offer this capacity for free to approved projects but remain gated by competitive access and national affiliation, while commercial clouds are globally accessible yet cost-sensitive. A globally equitable ecosystem would combine open research data sets with subsidized compute near data, ensuring that any researcher, anywhere, could run massive workflows without prohibitive costs or technical barriers.

To conclude, the heavy lift of the peta-scale era for seismological data analysis is within reach. Embracing cloud and other scalable computing solutions transforms this burden into a catalyst. It allows researchers to interrogate Earth processes at resolutions and timescales that were previously out of reach, opening new frontiers of discovery. It also permits researchers to focus on fundamental physical methods, rather than being limited by a given observational period and spatial extent. Petabyte-scale seismology is now practical.

ACKNOWLEDGMENTS

This work is supported by the Seismic Computational Platform for Empowering Discovery (SCOPED) project under the National Science Foundation (award numbers OAC-2103701 (UW), OAC-2103494 (UT)). The Schmidt Futures Foundation also supported the development of NoisePy at the University of Washington's Scientific Software Engineering Center. The EarthScope Consortium, through a Pass-Through Entity (PTE) Federal award no 2310069, partially supported this work. EarthScope data were accessed from the NSF SAGE data archive operated by the EarthScope Consortium (award number 1724509). The computing resources presented in this paper were obtained using CloudBank (Norman *et al.* 2021), which is supported by the National Science Foundation (award number CNS-1925001). The Harvard Data Science Initiative supported the development of the Julia Cloud workflow, developed by T Clements and J Schmitt, NSF EAR-1850015 award. JM has been funded by the European Union under the grant agreement n°101104996 ('DECODE'). We are grateful for discussions with Manochehr Bahavar and Loïc Bachelot surrounding visualization on the cloud. We thank Dr Andreas Fichtner and another anonymous reviewer for their constructive comments. AI tools were used for grammar and spelling checks.

DATA AVAILABILITY

The SCEDC and NCEDC data used in this study are publicly available through the AWS Open Data Sponsorship Program (https://registry.opendata.aws/southern-california-earthquakes/ for SCEDC and https://registry.opendata.aws/northern-california-earthquakes/ for NCEDC). The EarthScope Consortium data could be accessed through the EarthScope FDSN web services. Softwares used in Section 5 are available at https://github.com/noisepy and https://github.com/seisSCOPED/quakeScope/.

REFERENCES

Aagaard, B.T. *et al.*, 2025. *2024 california community earth models for seismic hazard assessments workshop report,* preprint(arXiv:2503.11545).

Abernathey, R.P. *et al.*, 2021. Cloud-native repositories for big scientific data, *Comput. Sci. Eng.,* **23**(2), 26–35.

Allen, B. *et al.*, 2012. Software as a service for data scientists, *Commun. ACM,* **55**(2), 81–88.

Arrowsmith, S.J., Trugman, D.T., MacCarthy, J., Bergen, K.J., Lumley, D. & Magnani, M.B., 2022. Big data seismology, *Rev. Geophys.,* **60**(2), e2021RG000769.

Bahavar, M. *et al.*, 2025a. *The cascadia region earthquake science center (crescent) community fault model viewer*. Version no. 1.0.1, Zenodo. doi:10.5281/zenodo.15092744.

Bahavar, M. *et al.*, 2025b. *The cascadia region earthquake science center (crescent) community velocity model viewer*. Version no. 1.0.1, Zenodo. doi:10.5281/zenodo.15092747.

Beckwith, R., 2011. Managing big data: Cloud computing and co-location centers, *J. Petrol. Technol.,* **63**(10), 42–45.

Beyreuther, M., Barsch, R., Krischer, L., Megies, T., Behr, Y. & Wassermann, J., 2010. Obspy: A python toolbox for seismology, *Seismol. Res. Lett.,* **81**(3), 530–533.

Breuer, A., Cui, Y. & Heinecke, A., 2019. Petaflop seismic simulations in the public cloud, in *International Conference on High Performance Computing,* pp. 167–185, Weiland, M., Juckeland, G., Trinitis, C. & Sadayappan, P., Springer.

Chen, P., Lee, E.-J. & Wang, L., 2013. A cloud-based synthetic seismogram generator implemented using windows azure, *Earthq. Sci.,* **26,** 321–329.

Clements, T. & Denolle, M.A., 2020. Seisnoise.jl: Ambient seismic noise cross correlation on the CPU and GPU in julia, *Seismol. Res. Lett.,* **92**(1), 517–527.

Clements, T. & Denolle, M.A., 2023. The seismic signature of california's earthquakes, droughts, and floods, *J. geophys. Res.: Solid Earth,* **128**(1), e2022JB025553.

Clements, T., Cochran, E.S., Baltay, A., Minson, S.E. & Yoon, C.E., 2024. Grapes: Earthquake early warning by passing seismic vectors through the grapevine, *Geophys. Res. Lett.,* **51**(9), e2023GL107389.

Clements, T., Schmitt, J.F. & Denolle, M.A., 2020. Cloud-native analysis of southern california waveform data, in *SCEC Annual Meeting, poster*.

Community, T.T.W., 2025. *The turing way: A handbook for reproducible, ethical and collaborative research*. Version no. 1.2.3, Zenodo. doi:10.5281/zenodo.15213042.

Consortium, C. *et al.*, 2018. *Cesium-an open-source javascript library for world-class 3D globes and maps*. https://cesium.com/platform/cesiumjs/?

Dancheva, T., Alonso, U. & Barton, M., 2024. Cloud benchmarking and performance analysis of an HPC application in amazon EC2, *Cluster Comput.,* **27**(2), 2273–2290.

Delph, J.R., Thomas, A.M. & Levander, A., 2021. Subcretionary tectonics: Linking variability in the expression of subduction along the cascadia forearc, *Earth planet. Sci. Lett.,* **556,** 116724.

Denolle, M.A. *et al.*, 2025. Training the next generation of seismologists: Delivering research-grade software education for cloud and HPC computing through diverse training modalities, *Seismol. Res. Lett.,* **96,** 3265–3279.

Feigl, K., 1969. PoroTomo Natural Laboratory Horizontal and Vertical Distributed Acoustic Sensing Data. doi:10.15121/1646880.

Fichtner, A., Ermert, L. & Gokhberg, A., 2017. Seismic noise correlation on heterogeneous supercomputers, *Seismol. Res. Lett.,* **88**(4), 1141–1145.

Gentemann, C.L., Holdgraf, C., Abernathey, R., Crichton, D., Colliander, J., Kearns, E.J., Panda, Y. & Signell, R.P., 2021. Science storms the cloud, *AGU Adv.,* **2**(2), e2020AV000354.

Glehman, J., Gabriel, A.-A., Ulrich, T., Ramos, M.D., Huang, Y. & Lindsey, E.O., 2024. *Partial ruptures governed by the complex interplay between geodetic slip deficit, rigidity, and pore fluid pressure in 3D cascadia dynamic rupture simulations,* EarthArXiv. doi:10.26443/seismica.v2i4.1427.

Gropp, W., Lusk, E., Doss, N. & Skjellum, A., 1996. A high-performance, portable implementation of the MPI message passing interface standard, *Parallel Comput.,* **22**(6), 789–828.

Guimarães, A., Lacalle, L., Rodamilans, C.B. & Borin, E., 2021. High-performance io for seismic processing on the cloud, *Concurr. Comput.: Practice Exp.,* **33**(18), e6250.

Habermann, T. *et al.*, 2021. Common data and metadata models for geophysical data in the cloud. *Authorea*. doi:10.1002/essoar.10509909.1.

Hauksson, E., Yoon, C., Yu, E., Andrews, J.R., Alvarez, M., Bhadha, R. & Thomas, V., 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**(4), 1961–1970.

Heinecke, A. *et al.*, 2014. Petascale high order dynamic rupture earthquake simulations on heterogeneous supercomputers, in *SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis,* pp. 3–14, IEEE.

Hutko, A.R., Bahavar, M., Trabant, C., Weekly, R.T., Fossen, M.V. & Ahern, T., 2017. Data products at the IRIS-DMC: Growth and usage, *Seismol. Res. Lett.,* **88**(3), 892–903.

Jiang, C. & Denolle, M.A., 2020. Noisepy: A new high-performance python tool for ambient-noise seismology, *Seismol. Res. Lett.,* **91**(3), 1853–1866.

Krauss, Z., Ni, Y., Henderson, S. & Denolle, M., 2023. Seismology in the cloud: guidance for the individual researcher, *Seismica,* **2**(2).

Krischer, L. *et al.*, 2016. An adaptable seismic data format, *Geophys. Suppl. Month. Notices R. Astron. Soc.,* **207**(2), 1003–1011.

Krischer, L. *et al.*, 2018. seismo-live: an educational online library of jupyter notebooks for seismology, *Seismol. Res. Lett.,* **89**(6), 2413–2419.

Kurtzer, G.M., Sochat, V. & Bauer, M.W., 2017. Singularity: Scientific containers for mobility of compute, *PloS One,* **12**(5), e0177459. doi:10.1371/journal.pone.0177459.

MacCarthy, J., Marcillo, O. & Trabant, C., 2019. Putting the cloud to work for seismology, *EOS, Trans. Am. geophys. Un.,* **100**(LA-UR-18-30252). doi:10.1029/2019EO119741.

MacCarthy, J., Marcillo, O. & Trabant, C., 2020. Seismology in the cloud: a new streaming workflow, *Seismol. Res. Lett.,* **91**(3), 1804–1812.

Makus, P. & Sens-Schönfelder, C., 2024. Seismic - an open source python toolset to compute velocity changes from ambient seismic noise, *Seismica,* **3**(1).

Martin, E.R., 2019. A scalable algorithm for cross-correlations of compressed ambient seismic noise, in *SEG International Exposition and Annual Meeting,* SEG, p. D043S141R005.

Melgar, D., Thomas, A.M., Sahakian, V.J., Meigs, A.J., Share, P.E., Tobin, H.J., Melbourne, T.I. & Elizabeth, J., 2024. *The Cascadia Region Earthquake Science Center Strategic Plan 2023-2027*. doi:10.5281/zenodo.11212220.

Merkel, D. *et al.*, 2014. Docker: lightweight linux containers for consistent development and deployment, *Linux J.,* **239**(2), 2. doi:10.5555/2600239.2600241.

Mkrtchyan, T. *et al.*, 2021. dcache: Inter-disciplinary storage system, in *EPJ Web of Conferences,* Vol. **251,** p. 02010, EDP Sciences.

Mohapatra, S., Yang, W., Yang, Z., Wang, C., Ma, J., Pavlis, G.L. & Wang, Y., 2025. Parallel seismic data processing performance with cloud-based storage. *Seismological Research Letters*. doi:10.1785/0220250115.

Mousavi, S.M., Ellsworth, W.L., Zhu, W., Chuang, L.Y. & Beroza, G.C., 2020. Earthquake transformer—an attentive deep-learning model for simultaneous earthquake detection and phase picking, *Nat. Commun.,* **11**(1), 3952.

Münchmeyer, J. *et al.*, 2022. Which picker fits my data? a quantitative evaluation of deep learning based seismic pickers, *J. geophys. Res.: Solid Earth,* **127**(1), e2021JB023499.

Münchmeyer, J., Bindi, D., Leser, U. & Tilmann, F., 2021. Earthquake magnitude and location estimation from real time seismic waveforms with a transformer network, *Geophys. J. Int.,* **226**(2), 1086–1104.

Ni, Y. *et al.*, 2025. *A global-scale database of seismic phases from cloud-based picking at petabyte scale,* preprint(arXiv:2505.18874).

Ni, Y., Denolle, M.A., Fatland, R., Alterman, N., Lipovsky, B.P. & Knuth, F., 2023. An object storage for distributed acoustic sensing, *Seismol. Res. Lett.,* **95**(1), 499–511.

Norman, M. *et al.*, 2021. Cloudbank: Managed services to simplify cloud access for computer science research and education, in *Practice and Experience in Advanced Research Computing 2021: Evolution Across All Dimensions, PEARC '21,* Association for Computing Machinery.

Peter, D. *et al.*, 2011. Forward and adjoint simulations of seismic wave propagation on fully unstructured hexahedral meshes, *Geophys. J. Int.,* **186**(2), 721–739.

Petersen, M.D. *et al.*, 2020. The 2018 update of the us national seismic hazard model: overview of model and implications, *Earthq. Spectra,* **36**(1), 5–41.

Pierleoni, P., Concetti, R., Belli, A., Palma, L., Marzorati, S. & Esposito, M., 2023. A cloud-IOT architecture for latency-aware localization in earthquake early warning, *Sensors,* **23**(20), 8431. doi:10.3390/s23208431.

Plesch, A., Marshall, S. & Shaw, J., 2024. *SCEC community fault model (CFM)*. Version no. 7.0, Statewide California Earthquake Center. doi:10.5281/zenodo.13685611.

Quinteros, J. *et al.*, 2021b. The GEOFON program in 2020, *Seismol. Res. Lett.,* **92**(3), 1610–1622.

Quinteros, J., Carter, J.A., Schaeffer, J., Trabant, C. & Pedersen, H.A., 2021a. Exploring approaches for large data in seismology: user and data repository perspectives, *Seismol. Res. Lett.,* **92**(3), 1531–1540.

Retailleau, L., Saurel, J., Zhu, W., Satriano, C., Beroza, G.C., Issartel, S., Boissier, P., Team, O. & Team, O., 2022. A wrapper to use a machine-learning-based algorithm for earthquake monitoring, *Seismol. Res. Lett.,* **93**(3), 1673–1682.

Schmitt, J., Clements, T. & Denolle, M., 2025. *Julians42/c4-project.jl: C4 project v0.1.0*.

Schmitt, J.F., Clements, T., Wang, N., Olsen, K.B. & Denolle, M.A., 2020. Ground motion prediction using ambient seismic noise on a large-n array in the La Basin., in *SCEC Annual Meeting*.

Seebeck, H. *et al.*, 2024. The New Zealand Community Fault Model–version 1.0: an improved geological foundation for seismic hazard modelling, *New Zealand J. Geol. Geophys.,* **67**(2), 209–229.

Sens-Schönfelder, C. & Wegler, U., 2011. Passive image interferometry for monitoring crustal changes with ambient seismic noise, *C. R. Geosci.,* **343**(8), 639–651.

Shapiro, N.M., Campillo, M., Stehly, L. & Ritzwoller, M.H., 2005. High-resolution surface-wave tomography from ambient seismic noise, *Science,* **307**(5715), 1615–1618.

Shaw, J.H. *et al.*, 2015. Unified structural representation of the Southern California crust and upper mantle, *Earth planet. Sci. Lett.,* **415,** 1–15.

Small, P. *et al.*, 2017. The SCEC unified community velocity model software framework, *Seismol. Res. Lett.,* **88**(6), 1539–1552.

Spica, Z.J. *et al.*, 2023. PubDAS: A public distributed acoustic sensing data sets repository for geosciences, *Seismol. Soc. Am.,* **94**(2A), 983–998.

Sun, H., Ross, Z.E., Zhu, W. & Azizzadenesheli, K., 2023. Phase neural operator for multi-station picking of seismic arrivals, *Geophys. Res. Lett.,* **50**(24), e2023GL106434.

The HDF Group, 1997–2023. *Hierarchical Data Format, Version 5,* https://www.hdfgroup.org/HDF5/.

Ventosa, S., Schimmel, M. & Stutzmann, E., 2019. Towards the processing of large data volumes with phase cross-correlation, *Seismol. Res. Lett.,* **90**(4), 1663–1669.

Walter, J.I., Ogwari, P., Thiel, A., Ferrer, F. & Woelfel, I., 2020. easyQuake: Putting machine learning to work for your regional seismic network or local earthquake study, *Seismol. Res. Lett.,* **92**(1), 555–563.

Wang, W., Wang, B. & Zheng, X., 2018. Public cloud computing for seismological research: calculating large-scale noise cross-correlations using aliyun, *Earthq. Sci.,* **31**(5–6), 227–233.

Wang, Y., Pavlis, G.L., Yang, W. & Ma, J., 2022. MsPASS: a data management and processing framework for seismology, *Seismol. Res. Lett.,* **93**(1), 426–434.

White, M.C., Zhang, Z., Bai, T., Qiu, H., Chang, H. & Nakata, N., 2023. HDF5eis: A storage and input/output solution for big multidimensional time series data from environmental sensors, *Geophysics,* **88**(3), F29–F38.

Wilkinson, M.D. *et al.*, 2016. The fair guiding principles for scientific data management and stewardship, *Sci. Data,* **3**(1), 1–9.

Witte, P.A., Louboutin, M., Modzelewski, H., Jones, C., Selvage, J. & Herrmann, F.J., 2020. An event-driven approach to serverless seismic imaging in the cloud, *IEEE Trans. Parallel Distributed Syst.,* **31**(9), 2032–2049.

Woollam, J. *et al.*, 2022. Seisbench—a toolbox for machine learning in seismology, *Seismol. Res. Lett.,* **93**(3), 1695–1709.

Wuestefeld, A. *et al.*, 2024. The global das month of February 2023, *Seismol. Res. Lett.,* **95**(3), 1569–1577.

Yoo, A.B., Jette, M.A. & Grondona, M., 2003. Slurm: Simple linux utility for resource management, in *Workshop on Job Scheduling Strategies for Parallel Processing,* pp. 44–60, Feitelson, D., Rudolph, L. & Schwiegelshohn, U., Springer.

Yu, E., Bhaskaran, A., Chen, S., Ross, Z.E., Hauksson, E. & Clayton, R.W., 2021. Southern california earthquake data now available in the aws cloud, *Seismol. Res. Lett.,* **92**(5), 3238–3247.

Zhan, Z., 2020. Distributed acoustic sensing turns fiber-optic cables into sensitive seismic antennas, *Seismol. Res. Lett.,* **91**(1), 1–15.

Zhang, M., Liu, M., Feng, T., Wang, R. & Zhu, W., 2022. Loc-flow: An end-to-end machine learning-based high-precision earthquake location workflow, *Seismol. Soc. Am.,* **93**(5), 2426–2438.

Zhou, J., Wei, Q., Wu, C. & Sun, G., 2021. A high performance computing method for noise cross-correlation functions of seismic data, in *2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom),* pp. 1179–1182, IEEE.

Zhu, W. & Beroza, G.C., 2019. Phasenet: a deep-neural-network-based seismic arrival-time picking method, *Geophys. J. Int.,* **216**(1), 261–273.

Zhu, W. *et al.*, 2025. *California earthquake data set for machine learning and cloud computing,* preprint(arXiv:2502.11500).

Zhu, W., Hou, A.B., Yang, R., Datta, A., Mousavi, S.M., Ellsworth, W.L. & Beroza, G.C., 2023. Quakeflow: a scalable machine-learning-based earthquake monitoring workflow with cloud computing, *Geophys. J. Int.,* **232**(1), 684–693.

Zhuang, J., Jacob, D.J., Lin, H., Lundgren, E.W., Yantosca, R.M., Gaya, J.F., Sulprizio, M.P. & Eastham, S.D., 2020. Enabling high-performance cloud computing for earth science modeling on over a thousand cores: Application to the geos-chem atmospheric chemistry model, *J. Adv. Model. Earth Syst.,* **12**(5), e2020MS002064.