Seismology in the Cloud: A New Streaming Workflow

Jonathan MacCarthy*[1], Omar Marcillo[1], and Chad Trabant[2]

Abstract

Data-intensive research in seismology is experiencing a recent boom, driven in part by large volumes of available data and advances in the growing field of data science. However, there are significant barriers to processing large data volumes, such as long retrieval times from data repositories, complex data management, and limited computational resources. New tools and platforms have reduced the barriers to entry for scientific cluster computing, including the maturation of the commercial cloud as an accessible instrument for research. In this work, we build a customized research cluster in the cloud to test a new workflow for large-scale seismic analysis, in which data are processed as a stream (retrieved on-the-fly and acted upon without storing), with data from the Incorporated Research Institutions for Seismology Data Management Center. We use this workflow to deploy a spectral peak detection algorithm over 5.6 TB of compressed continuous seismic data from 2074 stations of the USArray Transportable Array EarthScope network. Using a 50-node cluster in the cloud, we completed the noise survey in 80 hr, with an average data throughput of 1.7 GB per minute. By varying cluster sizes, we find the scaling of our analysis to be sublinear, due to a combination of algorithmic limitations and data center response times. The cloud-based streaming workflow represents an order-of-magnitude increase in acquisition and processing speed compared to a traditional download-store-process workflow, and offers the additional benefits of employing a flexible, accessible, and widely used computing architecture. It is limited, however, due to its reliance on Internet transfer speeds and data center service capacity, and may not work well for repeated analyses or those for which even higher data throughputs are needed. These research applications will require a new class of cloud-native approaches in which both data and analysis are in the cloud.

Cite this article as MacCarthy, J., O. Marcillo, and C. Trabant (2020). Seismology in the Cloud: A New Streaming Workflow, *Seismol. Res. Lett.* **XX**, 1–9, doi: 10.1785/0220190357.

Introduction

Research in seismology that requires tools and resources beyond those most commonly used due to large data volumes ("data-intensive" research) is experiencing renewed attention in the last decade, driven in part by advances in the new field of data science (e.g., Kong *et al.*, 2018; Bergen *et al.*, 2019) and by exponentially growing repositories of geophysical data at community data centers such as the Incorporated Research Institutions for Seismology (IRIS) Data Management Center (DMC) and the Observatories and Research Facilities for European Seismology (ORFEUS). Geographically large seismic deployments such as the USArray Transportable Array (TA) and AlpArray (Hetényi *et al.*, 2018), new dense deployments ("large-N") such as the Community Wavefields Demonstration Experiment (Sweet *et al.*, 2018), and high-sample-rate distributed acoustic sensing (DAS, Lindsey *et al.*, 2017) are capable of producing orders of magnitude more data per year than traditional broadband passive deployments. At the same time, established and emerging data-hungry analyses such as ambient noise tomography, waveform similarity detection, and neural networks and deep learning (Perol *et al.*, 2018; Ross *et al.*, 2018) are poised to provide denser geographic and temporal insights than has previously been possible.

The efficacy of these emerging techniques relies on the ability to access and process large volumes of data easily, yet doing so is becoming increasingly cumbersome under the conventional download-store-process workflow. A research effort will commonly start with prototyping an analysis using data from a limited time span and/or geographic region (prototype scale). As an algorithm is tested and validated, it may then be deployed over longer time periods or larger regions (experiment scale). If successful, a logical next step for many analyses is to expand the scale even further to find new times or regions for

1. Los Alamos National Laboratory, Los Alamos, New Mexico U.S.A.; 2. Incorporated Research Institutions for Seismology (IRIS) Data Services, Seattle, Washington, U.S.A.

*Corresponding author: jkmacc@lanl.gov

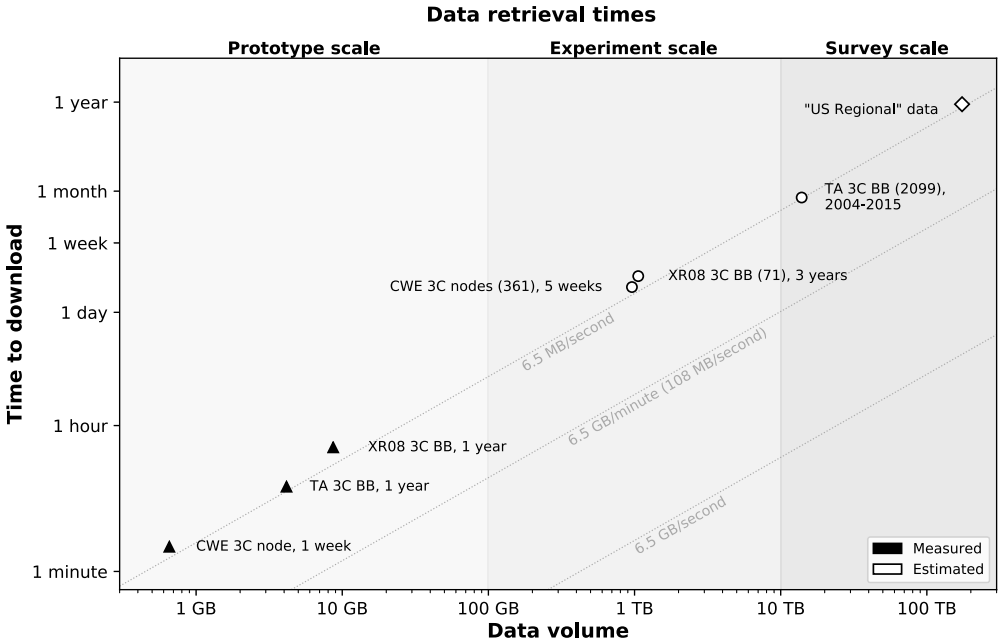

Figure 1. Data volumes and retrieval times from the Incorporated Research Institutions for Seismology (IRIS) Data Management Center (DMC) over several scales. Data for filled symbols were downloaded with ObsPy's FDSN Web Services "mass_downloader" using three concurrent threads, profiled, and used to estimate retrieval times for unfilled symbols. The Community Wavefields Demonstration Experiment (CWE) is 5 weeks of 361 three-component (3C) nodal stations sampled at 250 samples per second, the Transportable Arrary (TA) is 2099 3C broadband (BB) stations sampled at 40 samples per second, XR08 is 2–3 yr of 71 3C BB stations sampled at 40 samples per second, and "US Regional" data volume is estimated from IRIS DMC Data Statistics webpage.

which the analysis is still successful as well as those for which it fails (survey scale), yet this step remains difficult to do.

At the experiment scale, research is traditionally performed through a combination of protracted data retrieval, local management, and parallel or high-performance computing (HPC), and each step comes with a growing cost in a large-data regime. To illustrate "retrieval cost," we measure and estimate retrieval times for datasets of various sizes from the IRIS DMC (Fig. 1). We used ObsPy's International Federation of Digital Seismograph Networks Web Services (FDSN-WS) "mass_downloader" on a traditional four-core workstation to download one year of continuous three-component (3C) broadband data from a TA station and a temporary deployment station (Seismic Investigation of Edge-Driven Convection Associated with the Rio Grande Rift, network code XR 2008-2010), as well as 1 week of 3C nodal seismometer data from the IRIS Community Wavefields Demonstration Experiment. We used the profiled download times and data volumes to estimate retrieval times for all data from the respective networks, using data volumes obtained from the IRIS DMC "availability" webservice. For example, one year of 3C 40 samples per second miniSEED data took over 10 min to download at the observed rate of 6.5 MB/s. This is acceptable for most "prototype scale" and "experiment scale" research problems, but becomes prohibitive for "survey scale" analysis. We estimate that all 3C broadband seismic data from the TA would take nearly a month to download, and "US Regional" data (from IRIS DMC Data Statistics, see Data and Resources) would take a year, assuming uninterrupted acquisition. This retrieval cost inhibits "survey scale" research and will continue to grow as high sample-rate instruments, such as nodal seismometers, and large research datasets become more common (Incorporated Research Institutions for Seismology, 2019).

After large volumes of data are downloaded, the researcher then assumes the role of the data center, and is responsible for storing, indexing, and querying the data in their own duplicate repository. This is commonly done using simple file naming conventions, a database management system, or a combination of new tools such as Apache Accumulo and Hadoop Distributed File System (Magana-Zook *et al.*, 2016; Junek *et al.*, 2017). Mitigating common problems such as data gaps, metadata management, and infrastructure maintenance can cause these systems to become complex, requiring skills orthogonal to research. Finally, calculations larger than can be in a reasonable time on a large workstation, or computing server have traditionally required access to HPC or other institutional clusters, and familiarity with their tools, such as message passing interface (Chen *et al.*, 2016) or Hadoop and Spark (Addair *et al.*, 2014; Magana-Zook *et al.*, 2016). This further limits such research to groups that have access to these systems and skills.

In this article, we describe a new workflow for large-scale seismic analysis that seeks to minimize the challenges outlined previously. We created a scientific computing cluster using Amazon Web Services (AWS) Elastic Compute Cloud (EC2) instead of using local computing servers or HPC, which accelerates both data access and processing. We use standardized web services (FDSN-WS), programmatic data access interfaces over HTTPS, to stream data directly from the IRIS DMC to our cluster in the cloud for on-the-fly processing without storing it first, avoiding data management challenges. Finally, we employ

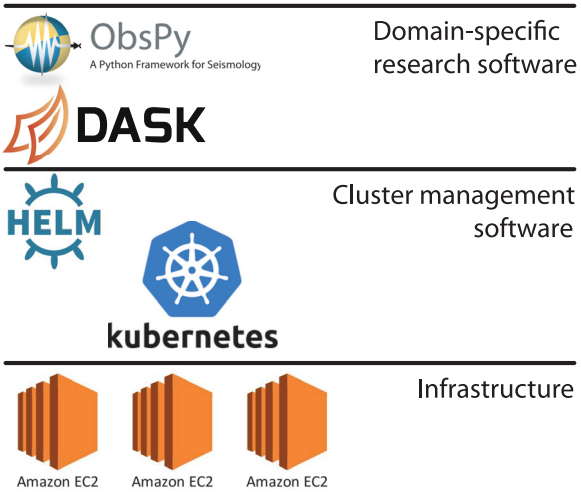

Figure 2. The technical architecture used in our cloud-based streaming workflow. We used Amazon EC2 to build the cluster, Kubernetes for cluster management, Helm for software installation, Dask for distributed parallelism in Python, and ObsPy for data acquisition and processing.

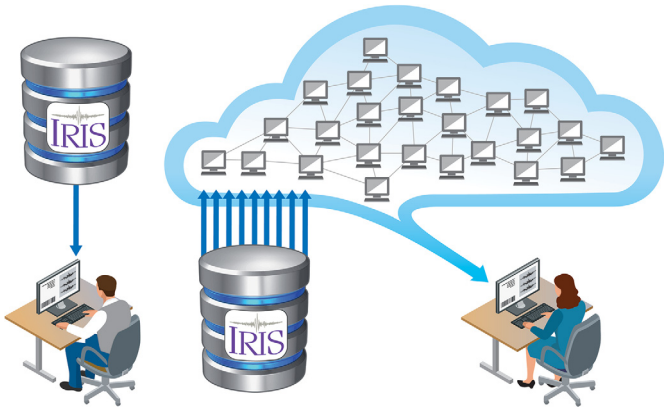

Figure 3. Traditional download-store-compute workflows (left) are limited by single-user data acquisition speed. A cloud-based streaming workflow (right) accelerates acquisition and computation time using many compute nodes acting as a swarm of users, each with their own IP address and coordinated workload.

new interactive parallel computation tools in Python to deploy a "survey scale" calculation using familiar desktop-scale research tools in the scientific Python ecosystem, reducing the learning curve for large-scale research. We demonstrate the workflow with a continent-scale survey of harmonic noise in continuous seismic data from the USArray TA (Marcillo and Carmichael, 2018; MacCarthy *et al.*, 2019; Marcillo and MacCarthy, 2020), and assess its scaling properties.

A Custom Research Cluster in the Cloud

Our cloud-based analysis comprises several layers: computing infrastructure, cluster management software, and domain-specific research software (Fig. 2). Commercial cloud providers such as Google Cloud Platform, AWS, or Microsoft Azure offer affordable (many price-points) and flexible (different hardware specifications) computing clusters through web-based or command-line interfaces. These have historically been used by companies to respond to diverse and time-varying business workloads, but the tools developed to make clusters accessible for commerce have also made them accessible for research. We chose AWS EC2 for its maturity and community knowledge online (see Data and Resources). The cluster consisted of 50 t2.large (2 CPU, 8 GB RAM) Debian Linux instances in the "us-west-2" region (Oregon), selected for its geographic proximity to the primary IRIS DMC in Seattle, Washington.

Cluster management software is needed to control the lifecycle and customization of a cluster. We use the Kubernetes orchestration system to provide and manage our EC2 node instances, due to its active community and support across multiple cloud providers, and the Helm cluster application manager to install our domain-specific research software. To parallelize our Python analysis code, we extend an existing Helm application that uses the distributed computation library, Dask (Dask Development Team, 2016). The application installs onto nodes in the cluster: Python, NumPy for array calculations (Oliphant, 2006), Pandas for tabular data manipulation (McKinney, 2011), a Dask distributed task scheduler and multiple Dask workers, and Jupyter for interactive analysis. To this stack we added ObsPy (Krischer *et al.*, 2015), the *de facto* seismic analysis toolbox for Python, and our own detection application. With these tools, a 50-node custom "do-it-yourself" research cluster was provisioned and configured for distributed seismic analysis in less than an hour.

One benefit of using cloud infrastructure is that it can be customized for the needs of the application. Our streaming workflow acquired data on-demand from the IRIS DMC, which required many small compute nodes with public IP addresses. Each worker requests a span of continuous data, performs a simple calculation, passes its results to the next available worker as directed by the Dask scheduler, and releases its memory without storing the waveforms (on-the-fly calculation). The DMC limits users by rate and concurrent count of connections per IP address. Each EC2 node is assigned its own public IP address, which makes the cluster appear to the DMC as a tightly coordinated swarm of users (Fig. 3). Ideally, as long as no single node exceeds the rate limits, the calculation can scale almost linearly with the number of nodes, until the hard limit of 100 IP addresses in the cluster routing table in AWS is reached or the DMC becomes saturated with requests. Subverting a data center's connection rate control mechanisms should only be done in coordination with the data center. Cluster configurations to maximize data throughput while

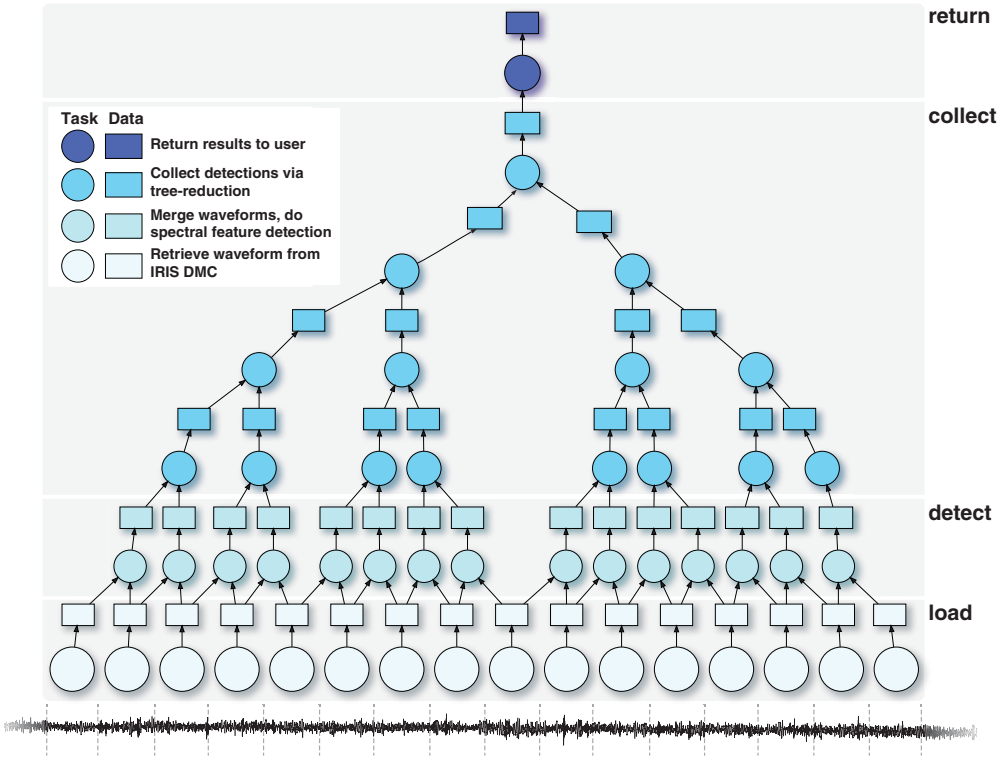

Figure 4. A computational graph representing a moving-window seismic detection application on two days of continuous data from a single channel. Circles represent tasks (Python functions), rectangles represent data consumed or produced by tasks, and arrows represent dependencies and the flow of the calculation. In our application, "load" tasks (bottom) request three hours of continuous data from the DMC, and are consumed two at a time by "detect" tasks. Results are collected in a binary tree reduction for efficiency, and returned to the user.

minimizing impact on DMC operations are being explored, but are outside of the scope of this article.

We choose the Python programming language to implement our application, as it is familiar to researchers, quick to learn, and contains a growing number of scientific libraries. ObsPy provided useful data structures and access to data at the DMC via a FDSN-WS client. The Dask distributed library provides a simple specification to describe, submit, and execute a computational graph on a distributed system (Rocklin, 2015; Dask Development Team, 2016). Figure 4 depicts a computational graph for a detection algorithm over two days of continuous single-channel seismic data. Tasks (Python functions and their associated inputs) at the base of the graph request different spans of data, which are passed to detection tasks, whose results are ultimately combined and returned to the researcher. Tasks in the graph are executed in dependency order, but tasks that load data and those that do calculations are executed concurrently, which distributes data requests throughout the lifespan of the calculation and reduces the request load on the DMC at any single time.

Application: Continent-Scale Feature Extraction and Detection Using Continuous Seismic Data

We test our cloud-based infrastructure with a continental-scale detection of harmonic tonal noise (TN), a signature of persistent mechanical energy sources (Marcillo and Carmichael, 2018). TN is a spectral feature characterized by discrete narrowband peaks and sequences of overtones. The TN detector consists of overlapping windowed Fourier transforms and detection of spectral peak sequences between ~0.5 and ~20 Hz. This is an example of an algorithm that was prototyped on a limited dataset (Marcillo *et al.*, 2015), expanded to a larger "experiment scale" application (Marcillo and Carmichael, 2018), and is now being used in a "survey scale" investigation (MacCarthy *et al.*, 2019; Marcillo and MacCarthy, 2020). Further, our TN application is just one example of a broad class of continuous-data feature extraction and detection applications, and our results may have broad applicability to other applications in this class.

Using our streaming workflow, we ran our TN detector over continuous seismic data for all BHZ channels of the _US-TA (USArray TA) virtual network available at the IRIS DMC, which consists of 2074 stations in 14 networks (see Data and Resources). Data were requested in three-hour increments and merged into six-hour moving windows with 50% overlap for detection. Over 4000 channel-years of data were retrieved (but not stored) in more than 12.7 million waveform requests. The total volume of seismic data analyzed was 5.6 TB, with an average data throughput of 1.7 GB/ min to the cluster during active processing. The streaming TN survey was completed in 3 days and 8 hours, during 8–11 February 2019. Figure 5 depicts the time progression of the survey, performed station-by-station from west to east. Stations in Alaska and the west coast were completed on the first day, including the deep archives (ten or more years of station data) of the CI network in southern California. The western United States was completed on the second day, the midwestern United States on

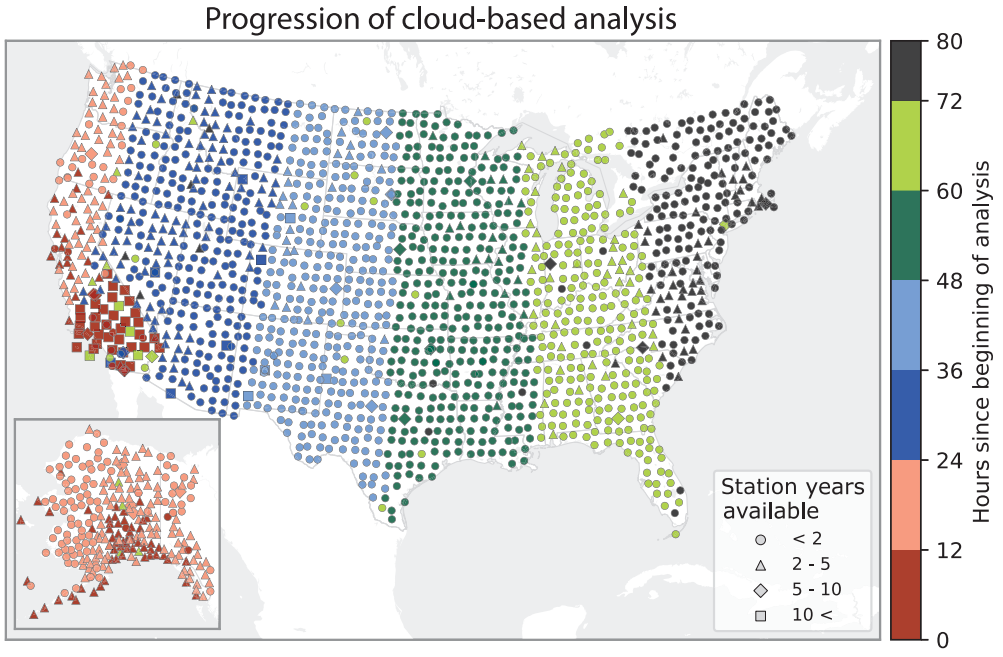

Figure 5. Map of stations processed in the industrial noise survey. Symbol colors correspond to the time a station was completed (processing west-to-east) and shapes to the number of station-years of data held at the IRIS DMC for that station. Inset map shows the region of Alaska.

the third, and the eastern United States and backfill (due to request errors) in the last eight hours. Mean station processing rate was approximately 15–20 s per station-year. The costs to AWS were ~$120 per day, almost entirely due to EC2 usage, resulting in a total cost of just over $400. The results of the survey are described in a companion article (Marcillo and MacCarthy, 2020).

Scalability of Streaming Analysis

The TN survey described previously was performed with the coordination of the IRIS DMC, so that the scalability and impact of streaming analyses on DMC operations could be assessed. These depend on many factors, including the size of the cluster and the performance of the algorithm, the distributed task scheduler, the data center, and the cloud provider. Although many of these factors are beyond our ability to quantify, we did examine how data throughput and data center connection rate vary with cluster size by performing time-limited TN detection runs (two hours each) using clusters of between 1 and 100 nodes, and analyzing server logs from the IRIS DMC covering this activity.

A striking increase in FDSN-WS connection rate at the DMC is associated with the onset of the 8–11 February TN survey (Fig. 6a). The mean connection rate originating from a 50-node cluster during the time-limited run is more than half of the entire normal community rate of ~4900 connections per minute during the period of the survey (Fig. 6b). As cluster size is increased from 1 to 100 nodes, connection rates from the cloud to the DMC further approach and sometimes exceed this background rate. The total amount of data processed for the two-hour runs on 1, 10, 20, 50, and 100-node clusters was 12, 102, 178, 195, and 207 GB, respectively. Because of the factors mentioned previously and discussed subsequently, data throughput was highly variable (Fig. 6c), and these values should be considered only approximately representative of cluster performance for other streaming applications of this class.

Both connection rate and data throughput rate scale sublinearly with cluster size, and vary more in the 50- and 100-node clusters than in the smaller ones. We attribute both to a combination of sublinear algorithm scaling, larger FDSN-WS response times for the two larger clusters, and variations in cloud provider performance. Stations were processed in series, and final execution of the computational graph for each station required the results of all three-hour segments. Delays in processing individual windows delayed completion of the entire graph. Although most FDSN-WS response times were much less than 10 s, some for the 50- and 100-node clusters were larger than 60 s, causing both the connection rate and data throughput to vary significantly. It is not clear whether the large AWS clusters directly impacted DMC response times or if they simply experienced a greater variety of response times through brute force sampling. It is clear, however, that our streaming analysis scales less than linearly with cluster size.

It is also clear that great care should be exercised to coordinate with the DMC or other data providers to undertake such a streaming analysis, so that operations are not negatively impacted and resources remain accessible to other users. As cloud computing becomes increasingly easier for researchers to leverage and the definition of "user" changes (i.e., a single IP address vs. a swarm of IP addresses), traditional methods to limit data center resource usage by a single "user" are less effective. Furthermore, the resource usage varies significantly by the style of the access (usually dictated by the study itself) and is difficult to generalize into guidelines. For example, large numbers of requests for small time windows of data encounter different limiting factors than fewer requests for larger selections of data. Finally, it is likely that cloud provider service speed varies over time, sometimes known as the "noisy neighbor"

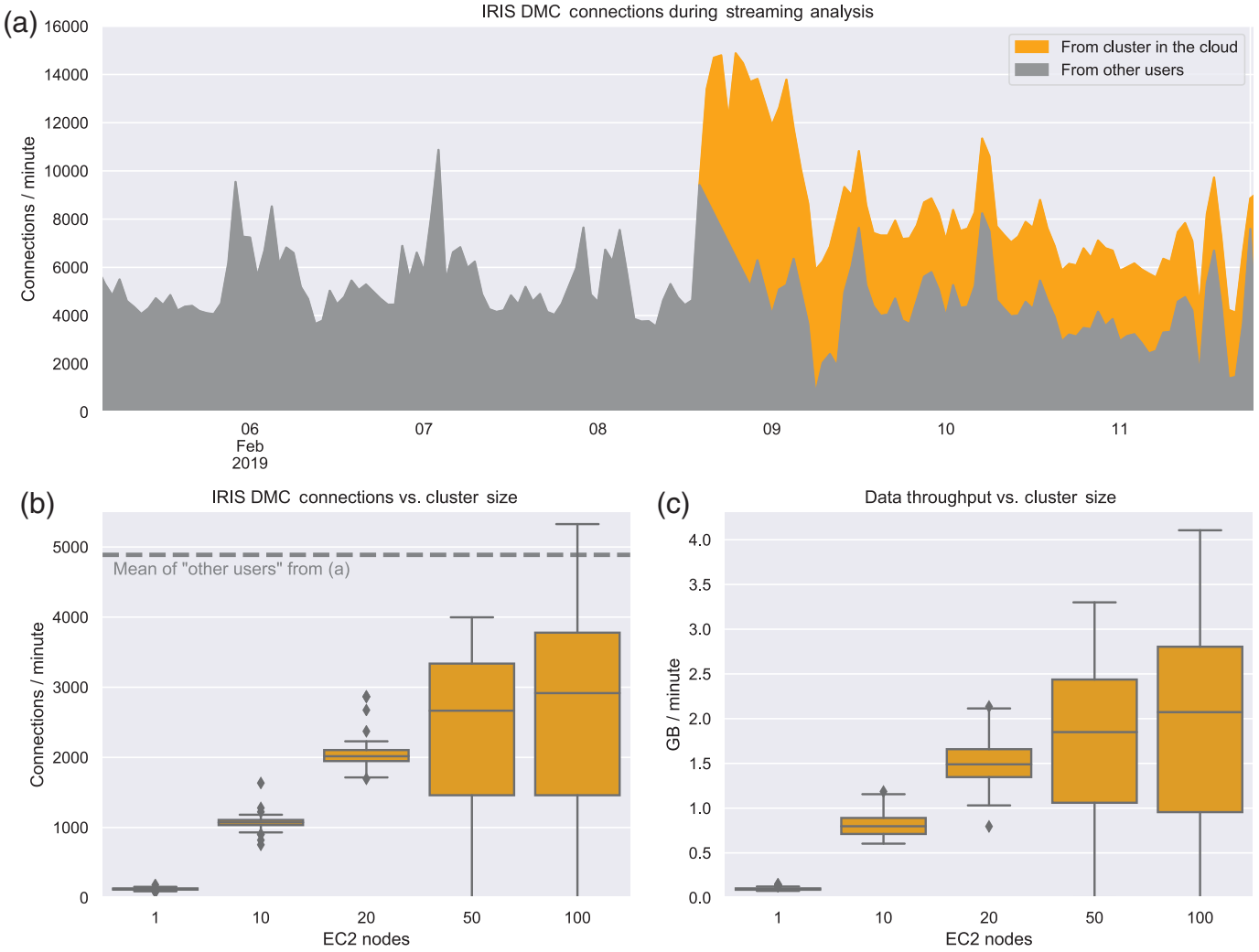

Figure 6. (a) Mean FDSN WebServices connections per minute received at all primary and auxiliary IRIS DMC servers during 5–12 February, 2019 (UTC), showing connections originating from the 50-node Amazon Web Services (AWS) cluster used in Figure 5 (orange) and those from normal community usage (gray). (b) Connection rate distributions for AWS clusters of varying sizes. Mean connection rate for normal DMC usage is calculated from (a). (c) Data ingestion rate from IRIS DMC to AWS for various cluster sizes.

problem, as EC2 instances are virtual computers running on shared hardware. This makes predicting performance challenging, so the scalability behavior described here should, again, be considered approximate. Although this work would benefit from additional tuning to improve throughput and reduce potential data center impact, the objective of the analysis was not to optimize this particular calculation, but instead to test the streaming workflow as simply as possible. For example, the three-hour request length we employed was chosen to simplify the following calculation, and may not have been the optimal length for a given data center to fulfill.

Discussion and Conclusions

The cloud-based streaming workflow demonstrated here offers two essential benefits over a traditional download-store-process workflow: scalability and flexibility. Using a cluster in the cloud and existing open-source software, we were able to acquire and process 5.6 TB of compressed seismic data, over 12.7 million continuous three-hour waveform segments, in 80 hr. This represents an order-of-magnitude increase in acquisition and processing speed compared to a traditional workflow. Further, the cloud infrastructure employed in our application was provisioned and instantiated by a single researcher using only a commercial cloud account without the need for access to traditional clusters or HPC systems. The ability to execute independently empowers researchers to rapidly iterate on a great variety (e.g., signal detection, ambient noise correlation, waveform feature extraction, or machine learning model training) and scale of analyses (e.g., local sparse to global dense datasets). Although there is a learning curve for working in the cloud, as with HPC, the consistent user experience, broad and loquacious user community, and autonomy

to experiment are a potent combination to make researchers quickly productive. Unrestricted by the administrative and security requirements that govern HPC systems nor the tedium of data storage, streaming continent-scale applications like ours can be executed in days instead of months. Finally, cloud infrastructures are flexible enough to adapt to the evolving needs of many research applications (e.g., processing speed, memory requirements, graphical processing unit vs. central processing unit architecture).

The streaming workflow is best suited for exploratory analyses on ephemeral computing resources. For repeated computations or for algorithms that require frequent data re-access, streaming directly from data centers becomes less favorable due to its reliance on data transfer over the Internet compared to the speed of fully in-cloud data storage. To illustrate, we compare the throughput achieved by our streaming calculation with a similar calculation performed using data stored in Amazon's Simple Storage Service (S3) and a cluster of the same size. The mean streaming throughput of our 100-node cluster was approximately 2 GB per minute, while that achieved using data stored in S3 was 43 GB per minute. At the streaming rate, the 5.6 TB of data in our application would be processed in approximately two days, while it would take just two hours at the in-cloud rate. Even more compelling is the fact that in-cloud processing is not limited by the number of nodes with public IP addresses, so processing could be further accelerated simply by adding more nodes. In-cloud storage will have a dramatic impact on the time it takes to access large data sets, such as the PoroTomo DAS dataset (~45 TB; Wang *et al.*, 2018), or the Caltech/U.S. Geological Survey Southern California Seismic Network (SCSN) continuous waveform archive, recently deployed to AWS (~100 TB; Hauksson *et al.*, 2020). Instead of weeks or months to retrieve at over-the-Internet speeds, these datasets could be retrieved in 18 and 38 hr, respectively, at in-cloud speeds like the one we achieved.

Another benefit of in-cloud storage is that data and analyses can be easily shared, for example, by setting the appropriate permissions in the storage service. With open permissions, Amazon S3 data can be accessed or downloaded by anyone, using a web browser or one of many programmatic interfaces. For large datasets, this can dramatically increase data usage and access speed, particularly for other in-cloud users. Further, on a common computing platform, actual calculations can be shared in the form of code repositories and Docker containers, for example, enabling new collaborative workflows based not just on shared data, but also on shared computing environments. Although costs may be incurred for data that are accessed or downloaded from outside of a cloud provider's platform, these costs can be managed, in the case of data in S3, by assigning it "Requester Pays" permissions. This type of access requires an AWS account that will be charged for any costs incurred. A full exploration of storage and access options for in-cloud data is outside of the scope of the work presented here, but is an area of ongoing investigation and is the subject of future work.

In addition to retrieval speed considerations, data center capacity is finite and commonly designed to support present-day usage patterns. Although spare capacity usually exists, widespread adoption of the streaming workflow presented here represents a pattern of significant increase in data access and would likely have negative impact on data center performance. This could not only reduce both streaming and traditional retrieval speeds, but also place a large operational burden on data centers to adjust to new access patterns. It is, therefore, strongly recommended to coordinate with data centers before executing a streaming calculation. To increase capacity and allow these in-cloud data access rates, the IRIS DMC is evaluating and working toward providing its data repositories and associated services within or near cloud or HPC systems. Also, the Southern California Earthquake Data Center has recently deployed its continuous waveform archive to AWS S3 storage, enabling in-cloud retrieval speeds and new access patterns like those described here (Hauksson *et al.*, 2020).

The cost of data storage and computation is important when considering working in the cloud. Each cloud service provider has a different fee structure and cost estimation tools (see Data and Resources). Using our streaming application, we describe our costs and illustrate a trade-off between storage and computation costs, with the caveat that each application will vary in its needs. No data were stored in our application, so cluster run time was the dominant cost. Our three-day calculation using 50 nodes cost over $400, and this cost would be incurred for each such streaming calculation. Depending on the data volume and number of repeated analyses, however, storing data in the cloud may offset this cost, through much shorter computation time. Presently, storing 5.6 TB in S3 costs only $130 per month, and the corresponding two-hour 100-node calculation using these data costs only $20 each time. With some information about these cost regimes, a detection survey like ours may be something that researchers may consider budgeting and planning for in their future proposals.

In addition to speed, data center service capacity and costs, there are a number of other practical challenges for cloud-based analyses. Most research software is not written for distributed systems, and most researchers are not familiar with the tools of working in the cloud. Error handling, monitoring, and security are more difficult on a distributed system compared to a desktop or large server. Finally, standard seismic formats, such as miniSEED, SAC, SEG-Y, or PH5 may not be optimal for access on distributed systems, where the balance between compression, file size, and network communication is different compared to local or HPC systems.

Although these challenges are daunting, working in the cloud also comes with a large base of community knowledge to overcome them, analogous to the advantages associated with

adopting a Portable Operating System Interface (POSIX) compliant operating system. This knowledge has been assembled by a wide and diverse community, including practitioners of both research and business. For example, the Dask parallelization library and the cluster management software Kubernetes and Helm were originally developed by corporations and data scientists: Anaconda Inc., Google, and Deis, respectively. Similarly, advances in cloud security and monitoring are constantly being made in the business community. New cloud-friendly compression algorithms such as Zstandard, first developed by Facebook, and storage formats such as Zarr, originally developed by Alistair Miles at Oxford University and the Centre for Genomics and Global Health, offer the opportunity for cloud-performant data access beyond traditional bespoke seismic formats (see Data and Resources). An example of how these disparate areas of expertise can add value to each other to address problems of scale in the earth science community is in the Pangeo project (Robinson *et al.*, 2019). We view the streaming workflow presented here as a bridge between traditional research approaches and those that fully adopt such cloud-native technologies. Though moving to the cloud requires a time investment, the benefits of working with a large and diverse community on common computing platforms may represent as great an opportunity for accelerating modern large-scale seismological research as the establishment of community instrument pools and data repositories. We see the adoption of the cloud as a viable and exciting option for data-intensive research in seismology.

Data and Resources

Waveform data used in this study were acquired from the Incorporated Research Institutions for Seismology (IRIS) Data Management Center (DMC). Networks providing data were AK (Alaska Earthquake Center, Univ. of Alaska Fairbanks, 1987), AT (NOAA National Oceanic and Atmospheric Administration (USA), 1967), AV (Alaska Volcano Observatory/USGS, 1988), AZ (Vernon and UC San Diego, 1982), BK (Northern California Earthquake Data Center, 2014), CI (California Institute of Technology and United States Geological Survey Pasadena, 1926), CN (Geological Survey of Canada, 1989), II (Scripps Institution of Oceanography, 1986), IU (Albuquerque Seismological Laboratory (ASL)/USGS, 1988), LD (The Lamont–Doherty Cooperative Seismographic Network), NN (University of Nevada, Reno, 1971), TA (IRIS Transportable Array, 2003), US (Albuquerque Seismological Laboratory (ASL)/USGS, 1990), and UU (University of Utah, 1962). Analysis was performed using Python 3.7, NumPy, SciPy, Pandas, ObsPy, and Dask. Plots were made using Matplotlib (Hunter, 2007) and Basemap. Amazon EC2 documentation can be found at https://docs.aws.amazon.com/ec2/index.html (last accessed October 2019). IRIS DMC Data Statistics retrieved from http://ds.iris.edu/data/distribution/ (last accessed February 2019). Cloud cost calculators can be found at https://calculator.s3.amazonaws.com/index.html (Amazon Web Services), https://azure.microsoft.com/en-us/pricing/calculator/ (Microsoft Azure), https://cloud.google.com/products/calculator/ (Google Cloud Platform), all last accessed October 2019. Announcement of Zstandard can be found at https://engineering.fb.com/core-data/smaller-and-faster-data-compression-with-zstandard, and documentation for Zarr at https://zarr.readthedocs.io/en/stable/ (both last accessed October 2019).

Acknowledgments

The authors would like to thank the Los Alamos Information Science & Technology Institute and the Office of the Chief Information Officer for initial support of this work. The authors also thank Robert Weekly, Robert Casey, and Inge Watson at the Incorporated Research Institutions for Seismology (IRIS) Data Management Center (DMC) for their collaboration during the experiment. This article has been authored by Triad National Security under Contract Number 89233218CNA000001 with the U.S. Department of Energy. The U.S. Government retains and the publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a nonexclusive, paidup, irrevocable, worldwide license to publish or reproduce the published form of this article, or allow others to do so, for U.S. Government purposes. It is approved for unlimited release as LA-UR-19-31275.

References

Addair, T. G., D. A. Dodge, W. R. Walter, and S. D. Ruppert (2014). Large-scale seismic signal analysis with Hadoop, *Comput. Geosci.* **66,** 145–154.

Alaska Earthquake Center, Univ. of Alaska Fairbanks (1987). Alaska Regional Network, International Federation of Digital Seismograph Networks. Dataset/Seismic Network, doi: 10.7914/SN/AK.

Alaska Volcano Observatory/USGS (1988). Alaska Volcano Observatory, International Federation of Digital Seismograph Networks. Dataset/Seismic Network, doi: 10.7914/SN/AV.

Albuquerque Seismological Laboratory (ASL)/USGS (1988). Global Seismograph Network (GSN - IRIS/USGS), International Federation of Digital Seismograph Networks. Dataset/Seismic Network, doi: 10.7914/SN/IU.

Albuquerque Seismological Laboratory (ASL)/USGS (1990). United States National Seismic Network, International Federation of Digital Seismograph Networks. Dataset/Seismic Network, doi: 10.7914/SN/US.

Bergen, K. J., T. Chen, and Z. Li (2019). Preface to the focus section on machine learning in seismology, *Seismol. Res. Lett.* **90,** no. 2A, 477–480.

California Institute of Technology and United States Geological Survey Pasadena (1926). Southern California Seismic Network, International Federation of Digital Seismograph Networks. Dataset/Seismic Network, doi: 10.7914/SN/CI.

Chen, P., N. J. Taylor, K. G. Dueker, I. S. Keifer, A. K. Wilson, C. L. McGuffy, C. G. Novitsky, A. J. Spears, and W. S. Holbrook (2016). pSIN: A scalable, Parallel algorithm for Seismic INterferometry of large-N ambient-noise data, *Comput. Geosci.* **93,** 88–95.

Dask Development Team (2016). *Dask: Library for Dynamic Task Scheduling*, http://dask.pydata.org (last accessed October 2019).

Geological Survey of Canada (1989). Canadian National Seismograph Network, International Federation of Digital Seismograph Networks. Dataset/Seismic Network, doi: 10.7914/SN/CN.

Hauksson, E., C. Yoon, E. Yu, J. R. Andrews, M. Alvarez, R. Bhadha, and V. Thomas (2020). Caltech/USGS Southern California Seismic

Network (SCSN) and Southern California Earthquake Data Center (SCEDC): Data availability for the 2019 Ridgecrest sequence, *Seismol. Res. Lett.*, doi: 10.1785/0220190290.

Hetényi, G., I. Molinari, J. Clinton, G. Bokelmann, I. Bondár, W. C. Crawford, J.-X. Dessa, C. Doubre, W. Friederich, F. Fuch, *et al.* (2018). The AlpArray seismic network: A large-scale European experiment to image the Alpine Orogen, *Surv. Geophys.* **39,** no. 5, 1009–1033.

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment, *Comput. Sci. Eng.* **9,** no. 3, 90–95.

Incorporated Research Institutions for Seismology (2019). IRIS PASSCAL to Expand Pool of Seismic Instruments, Retrieved from https://www.iris.edu/hq/news/story/iris_passcal_to_expand_pool_of_seismic_instruments (last accessed October 2019).

IRIS Transportable Array (2003). USArray Transportable Array, International Federation of Digital Seismograph Networks. Dataset/Seismic Network, doi: 10.7914/SN/TA.

Junek, W. N., C. A. Houchin, J. A Wehlen, J. E. Highcock, and M. Waineo (2017). Acquisition of seismic, hydroacoustic, and infrasonic data with Hadoop and Accumulo, *Seismol. Res. Lett.* **88,** no. 6, 1553–1559.

Kong, Q., D. T. Trugman, Z. E. Ross, M. J. Bianco, B. J. Meade, and P. Gerstoft (2018). Machine learning in seismology: Turning data into insights, *Seismol. Res. Lett.*, doi: 10.1785/0220180259.

Krischer, L., T. Megies, R. Barsch, M. Beyreuther, T. Lecocq, C. Caudron, and J. Wassermann (2015). ObsPy: A bridge for seismology into the scientific Python ecosystem, *Comput. Sci. Discov.* **8,** no. 1, 014003.

Lindsey, N. J., E. R. Martin, D. S. Dreger, B. Freifeld, S. Cole, S. R. James, B. L. Biondi, and J. B. Ajo-Franklin (2017). Fiber-optic network observations of earthquake wavefields, *Geophys. Res. Lett.* **44,** no. 23, 11–792.

MacCarthy, J., O. Marcillo, and C. Trabant (2019). Putting the cloud to work for seismology, *Eos Trans. AGU* **100,** doi: 10.1029/2019EO119741.

Magana-Zook, S., J. M. Gaylord, D. R. Knapp, D. A. Dodge, and S. D. Ruppert (2016). Large-scale seismic waveform quality metric calculation using Hadoop, *Comput. Geosci.* **94,** 18–30.

Marcillo, O., and J. MacCarthy (2020). Mapping seismic tonal noise in the contiguous US, *Seismol. Res. Lett.*, doi: 10.1785/0220190355.

Marcillo, O. E., and J. Carmichael (2018). The detection of wind-turbine noise in seismic records, *Seismol. Res. Lett.* **89,** no. 5, 1826, doi: 10.1785/0220170271.

Marcillo, O., S. Arrowsmith, P. Blom, and K. Jones (2015). On infrasound generated by wind farms and its propagation in low-altitude tropospheric waveguides, *J. Geophys. Res.* **120,** no. 19, 9855–9868.

McKinney, W. (2011). pandas: A foundational Python library for data analysis and statistics, *Python for High Performance and Scientific Computing*, 14.

NOAA National Oceanic and Atmospheric Administration (USA) (1967). National Tsunami Warning Center Alaska Seismic Network, International Federation of Digital Seismograph Networks. Dataset/Seismic Network, doi: 10.7914/SN/AT.

Northern California Earthquake Data Center (2014). Berkeley Digital Seismic Network (BDSN) [Data set], Northern California Earthquake Data Center, doi: 10.7932/BDSN.

Oliphant, T. E. (2006). *A Guide to NumPy*, Vol. 1, Trelgol Publishing, U.S.A., p. 85.

Perol, T., M. Gharbi, and M. Denolle (2018). Convolutional neural network for earthquake detection and location, *Sci. Adv.* **4,** no. 2, e1700578.

Robinson, N. H., J. Hamman, and R. Abernathey (2019). Science needs to rethink how it interacts with big data: Five principles for effective scientific big data systems, arXiv preprint arXiv: 1908.03356.

Rocklin, M. (2015). Dask: Parallel computation with blocked algorithms and task scheduling, in K. Huff and J. Bergstra (Editors), *Proc. of the 14th Python in Science Conference*, 130–136.

Ross, Z. E., M. A. Meier, E. Hauksson, and T. H. Heaton (2018). Generalized seismic phase detection with deep learning, *Bull. Seismol. Soc. Am.* **108,** no. 5A, 2894–2901.

Scripps Institution of Oceanography (1986). IRIS/IDA Seismic Network, International Federation of Digital Seismograph Networks. Dataset/Seismic Network, doi: 10.7914/SN/II.

Sweet, J. R., K. R. Anderson, S. Bilek, M. Brudzinski, X. Chen, H. DeShon, C. Hayward, M. Karplus, K. Keranen, C. Langston, and F. C. Lin (2018). A community experiment to record the full seismic wavefield in Oklahoma, *Seismol. Res. Lett.* **89,** no. 5, 1923–1930.

University of Nevada, Reno (1971). Nevada Seismic Network, International Federation of Digital Seismograph Networks. Dataset/Seismic Network, doi: 10.7914/SN/NN.

University of Utah (1962). University of Utah Regional Seismic Network, International Federation of Digital Seismograph Networks. Dataset/Seismic Network, doi: 10.7914/SN/UU.

Vernon, F., and UC San Diego (1982). ANZA Regional Network, International Federation of Digital Seismograph Networks. Dataset/Seismic Network, doi: 10.7914/SN/AZ.

Wang, H. F., X. Zeng, D. E. Miller, D. Fratta, K. L. Feigl, C. H. Thurber, and R. J. Mellors (2018). Ground motion response to an M_L 4.3 earthquake using co-located distributed acoustic sensing and seismometer arrays, *Geophys. J. Int.* **213,** no. 3, 2020–2036.

Manuscript received 22 November 2019
Published online 18 March 2020