Practical Analysis of Big Acoustic Sensor Data for Environmental Monitoring

Anthony Truskinger, Mark Cottman-Fields, Philip Eichinski, Michael Towsey, Paul Roe
QUT Ecoacoustics Research Group
School of Electrical Engineering and Computer Science
Queensland University of Technology
Brisbane, Australia
{anthony.truskinger, m.cottman-fields, phil.eichinski}@student.qut.edu.au, {m.towsey, p.roe}@qut.edu.au

***Abstract*—Monitoring the environment with acoustic sensors is an effective method for understanding changes in ecosystems. Through extensive monitoring, large-scale, ecologically relevant, datasets can be produced that can inform environmental policy. The collection of acoustic sensor data is a solved problem; the current challenge is the management and analysis of raw audio data to produce useful datasets for ecologists.**

This paper presents the applied research we use to analyze big acoustic datasets. Its core contribution is the presentation of practical large-scale acoustic data analysis methodologies. We describe details of the data workflows we use to provide both citizen scientists and researchers practical access to large volumes of ecoacoustic data. Finally, we propose a work in progress large-scale architecture for analysis driven by a hybrid *cloud-and-local* production-grade website.

Keywords—acoustic sensing; bioacoustics; data analysis; scalable analysis; cloud infrastructure; ecoacoustics

I. INTRODUCTION

Sensors are an effective tool for the large scale monitoring of the environment. Acoustic sensors are regularly used to monitor vocalizing fauna with the intent of assessing biodiversity [1, 2]. Acoustic sensor data can also address ecological questions relating to the vocalizing patterns of fauna, the presence or absence of species, and species abundance. The volume of data generated by sensors requires large compute resources for analysis. This paper elucidates the practical analysis methodologies that will allow for a hybrid *cloud-and-local* compute architecture required by our ecoacoustics project.

Traditional methods of surveying ecosystems are manual and require field workers to visit the site of study. While the results of manual surveys remain valuable, sensors have several advantages: they record data constantly, cost relatively little, are minimally invasive, and create a permanent, objective record of a site. Deploying sensors over large spatiotemporal scales allows scientists to collect massive amounts of data.

Advances in sensor technology, specifically in storage capacity, in the last 10 years, have provided the hardware for practical large-scale collection of data. The Wildlife Acoustics' SM2+ [3] is a commonly used acoustic sensor [4-7] that can be deployed with four high density SDHC cards and an external power supply. A solar-powered SM2+ sensor can record audio for over a year (128kbps MP3, 1024GB storage). With reliable sensors and high-density storage, collecting data is no longer considered problematic. Instead, ecoacoustics research now concentrates on the questions of managing and analyzing ecoacoustic data; the latter of which is a more complex and varied problem [8].

Automated methods of analyzing acoustic data are preferred; however, currently there exists no single, generalized, automated solution for identifying all vocalizing fauna within sensor audio recordings. There are two broad reasons for this intractability. First, automated identification of species is difficult due to the variability that faunal vocalizations exhibit, the low *signal to noise ratios* (SNR) endemic to acoustic sensors, and the acoustic competition between species that adds further complexity to the data [1]. Second, practical methods for analyzing, visualizing, and understanding acoustic sensor data are still not well developed. Raw audio data is opaque and hard to reason about without analysis [9, 10].

Analysis and management of ecoacoustics is a big data problem and our research to solve this problem has produced software artifacts such as the Ecosounds Acoustic Workbench (pictured in Fig 1). Employing the 5Vs of big data [11-13] as metrics, the QUT Ecoacoustics Research Group collects data that has:

- Volume: Currently, 24TB of acoustic sensor data has been collected. Of that, 15TB has been ingested into the Bioacoustic Workbench – a production website – where audio can be accessed (navigated, played, and shown as spectrograms) on demand.
- Velocity: The research group has access to 50 sensors; there is a potential data velocity of 355GB/day (Stereo WAVE, 22050Hz, 16-bit samples).
- Variety: While sensors produce data in consistent formats, the content can vary wildly over small geographical distances. Techniques applicable to one region often do not work in others. Additionally, various methods of analysis produce many types of data, including visualizations, indices, events, points of interest, spectra, metadata, annotations, or tags. Processes that involve people performing analysis can introduce further variety.

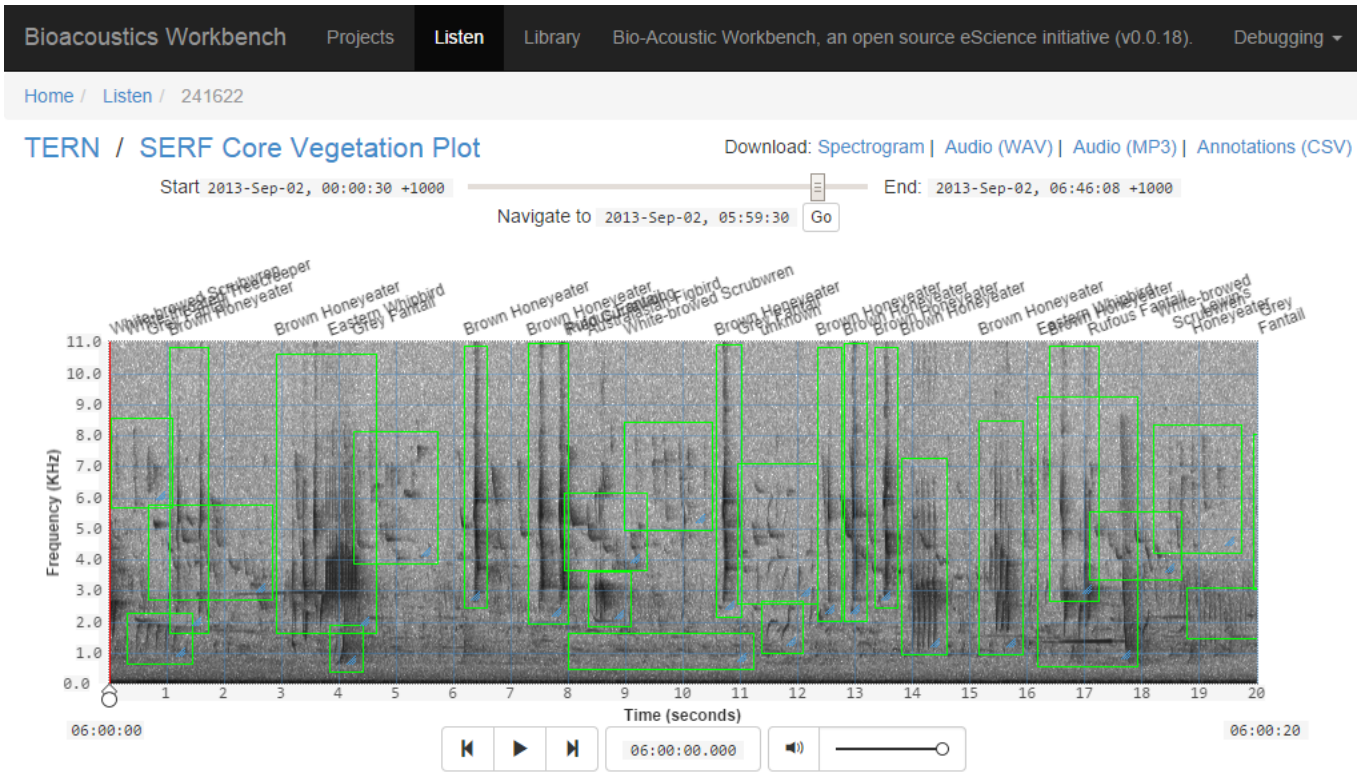

Fig. 1. A screenshot of the Ecosounds Bioacoustic Workbench's annotation interface

- Veracity: The raw data produced by sensors are an objective record of activity – this is an inherent advantage of using sensors over manual studies. However, human-driven analysis or the verification of automated analysis creates potential sources of data uncertainty.
- Value: The results from collecting and analyzing acoustic sensor data can produce valuable ecological data for input into the formation of environmental policies.

This paper presents software, methodologies, and supporting architecture for analyzing large sets of acoustic sensor data. Scientists within our research group and external collaborators have made use of the processes and software described by this paper. Our contribution is to publish our applied large-scale analysis research, details of our migration to cloud based architecture, and our open source software to aid other researchers in the field. Related work is presented, followed by an overview of the acoustic sensor data workflow. Then, a detailed report on methodologies is presented. Finally, a work in progress section details plans for scaling up the analysis architecture.

II. Related Work

There are a growing number of data intensive projects with varying research foci. Within data-intensive science, there are recognized differences in dataset sizes, computational needs, and collaboration standards. Our work is firmly in the middle of Jim Gray's long tail of science [11]. Large-scale ecoacoustics requires reasonably complex technology, as well as computer scientists and IT experts to manage and process data [14]. The volume of data being processed necessitates an evolution beyond spreadsheets, flat files, and hand-curated data – the methods of independent scientists.

While most audio datasets are not equivalent in size to genome or astronomy data (typically in the petabyte range) [15], terabytes of audio still pose a significant challenge. Volume on disk does not necessarily equate to complexity in processing. Acoustic data is opaque and by definition always represents data over time. This makes it difficult to summarize, visualize, or even manually preview individual files [10]. Effectively characterizing local areas as well as large amounts of data, obtained across large spatiotemporal periods, is challenging. Analysis of acoustic data using indices and broad methods of comparison and differentiation have been used to successfully obtain an overview for comparing acoustically similar areas [4].

Recordings of fauna vocalizing are commonplace. However, there is an important distinction to be drawn between targeted recordings and untargeted recordings. Targeted recordings, also known as trophy recordings, are usually short, contain just one call, have a high SNR, and are usually captured with specialized equipment. These recordings have a relatively low cost in terms of data volume and analysis complexity. Untargeted or general environment recordings, like those produced by acoustic sensors, are typically very long (hours to days per recording), have many vocalizing fauna, low SNRs, and can capture overwhelming amounts of irrelevant signal and

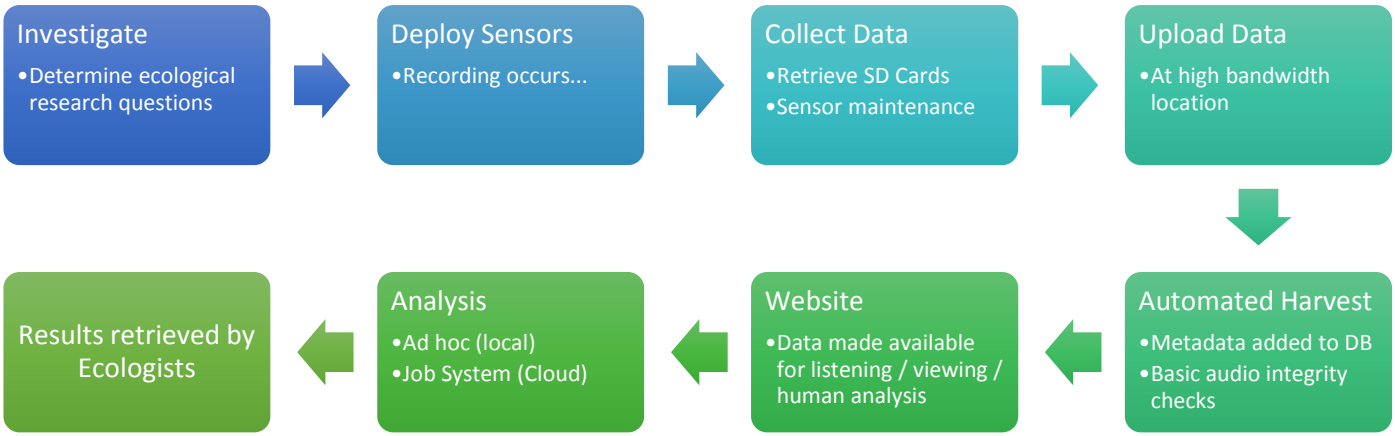

Fig. 2. The QUT Ecoacoustics Research Group's process for collecting data from sensors

background noise. These recordings have a high cost in terms of volume of data and analysis.

The Xeno Canto website is a collection of faunal vocalizations in targeted recordings. The majority of recordings are short, with a high SNR. The site has similar goals to our project – increasing the data available on the environment and biodiversity – with a vastly different approach. The short recordings lend themselves to manual listening and analysis. It is possible to discuss an entire recording and often be sure of which sound source is the 'target' of the recording. Xeno canto currently has approximately 500GB of audio recordings [16]. Sensors however, generate very large, untargeted, recordings – it is not feasible to discuss or analyze that data with Xeno Canto methods.

There are a number of commercial programs that can be used to analyze acoustic sensor data to detect vocalizations of interest. SongScope and Raven are two programs that can achieve reasonable accuracy in smaller audio datasets with supervised training [17]. Unfortunately, neither of these programs are designed to scale to very large datasets.

Pumilio is a successful open source ecoacoustics web application [18]. It has multiple deployments actively used by different research groups, allows for uploading, listening, and analyzing audio. The project has focused on easy deployment and use. Pumilio is designed to run on a single machine – possibly in the cloud – it is not clear how the project will deal with significant scale.

III. Methods – Data Collection

This section details the methods employed to gather acoustic sensor data by our group. This process is depicted by Fig 2.

Initially, ecological research questions are provided by collaborating ecologists, community environment groups, businesses concerned about their impact on the environment, or government initiatives. The research questions utilize acoustic information from sensors, sometimes indirectly, to form conclusions.

Sensors are deployed into the field in different configurations. Typically, recorders are placed at ecotones (sites that are a transition between two biomes) to maximize the variety of species detected. Sensors can also be deployed to target specific species or in patterns (like grids). Factors that affect sensor performance include territory size of targeted fauna, vocalization amplitude & frequency of target fauna, vegetation type, terrain, and environmental noise sources.

SM2+ sensors (Fig 3.) are the most commonly used; they can potentially record audio unattended for over a year. However, we typically employ one of two patterns: weeklong or four-month long cycles (deployed for up to 3 years). These shorter cycle times allow data to be incrementally gathered. When the data is gathered, health checks and maintenance are also conducted. Weeklong cycles require four D-cell batteries, whereas the four-month cycles (≈125 days) are deployed with a solar panel and a deep-cycle battery. Both types of deployment record data in a stereo WAVE format (PCM, 22050Hz, 16-bit samples). The SM2s have two microphone inputs – utilizing

Fig. 3. A deployed SM2+ Sensor

both microphones creates redundancy in the event of a single microphone failure.

At the end of a cycle, a field worker will inspect a deployed sensor. If it is the end of the deployment, the sensor is retrieved. If a deployment has not concluded, the SD cards are swapped out. Regardless, the cards are physically returned to a high bandwidth location (typically within a university's network) and the data is uploaded to a working area. When metadata files are added to each directory, an automated harvester detects the changes and schedules harvest jobs for each waiting audio file. Files are converted from WAC if necessary to WAVE – other file formats do not require pre-ingestion conversion. The file type *WAVE* is used for uncompressed files and *WAC* is Wildlife Acoustic's proprietary lossless audio compression format.

Required analyses, either automatic or semi-automatic, are conducted before the results are sent off to ecologists. Semi-automated analysis is done by annotating faunal vocalizations [1].

IV. Methods – Analysis Development and Execution

We are an eScience research group. Our goal is to provide computer science support to traditional scientists. Nevertheless, even within our group we hire/require specialist IT professionals in addition to research staff. We propose that the concept of eScience requires graduated levels of professional IT support for data intensive science; some groups may only need small amounts of professional support, others may need small workforces (e.g. the Square Kilometer Array project [19]).

A. Developer / Researcher Tension

There is tension between the goals of researchers and software developers. As an eScience group, we regularly work with research and professional staff. One core goal of the research group is to incorporate analysis algorithms and processes into the public production website. This requires a reasonable understanding of the source code and a fixed feature base. Contrast this with the typical methodology for research work: researchers are never done improving their results and are constantly tweaking source code. Without freezing core features and APIs, it is difficult to maintain working production code [20, 21].

We have approached this problem in two main ways: Refactoring checkpoints (freeze feature sets that researchers have stopped working on) and ad hoc analysis systems.

The first concept, freezing features is a common practice in software development. In order to ship a product, new features will not be allowed, existing features will have their APIs frozen, and the only continuing work will be maintenance. A full feature freeze is not compatible with a researcher's set of priorities.

As an alternative, every few months, time is allocated for refactoring analysis code. Features and APIs that have not changed recently are marked as 'production stable' and can then be depended on. Features that are part of active research are tracked but not altered. The result is a limited but progressive set of restrictions to the researchers. This semi-regular iteration cycle works well because all parties involved know and have input into the process. The result is a naturally forming framework that adapts as analysis algorithms are developed, tested, and become stable.

The second concept we have employed is ad-hoc analysis systems, which have proven very useful. We have reserved dedicated compute resources and have some generalized scripts for running ad hoc analyses. These scripts require an IT professional to run but do not require production-level feature freeze.

B. Compute Resources

We have three basic compute resources available:

- QUT's High Performance Computing (HPC) support
- a dedicated big data processing lab (BigData) containing powerful standalone computers designed for researcher experimentation
- Queensland Cyber Infrastructure Foundation (QCIF) and the National eResearch Collaboration Tools and Resources (NeCTAR) provide access to cloud storage and cloud compute resources for data-driven collaborative research.

Our research group currently has two storage options with 100TB in total through the QUT HPC and QCIF. The two storage locations have mirrors of all audio data. In addition to serving as backups, it allows either QCIF Cloud or QUT HPC compute resources to run analysis with on-site data access. We would prefer solutions that remove the need to transfer data [22]; however we currently remain dependent on high-speed links between data stores.

The transfer of data that involves disk or network I/O generally has the largest impact on analysis efficiency. The main method we employ to reduce the required data transfer is command-line audio manipulation tools that can seek smartly through audio files. For example, *mp3splt* can segment MP3 format files without needing to read the entire file. Early in the research group's development of analyses, the amount of data stored in RAM caused paging and extreme contention for resources. This limitation has been bypassed through audio file segmenting.

The next most limiting factor is the number of processing cores. A 'big data' lab provided by the university contains twelve machines (dual Intel Xeon E5-2665, 32 virtual cores, 256GB DDR3 RAM, 3TB SCSI Raid, dual 1Gb Ethernet) designed to address the needs of researchers working with data that is impractical to process on their personal computers. Their prime benefit to our research group is unrestricted access and resulting flexibility. We also make use of their high throughput and large amount of RAM. In particular, RAM disks for storing the cache of intermediate audio files cut for each segment of analysis are very useful.

Similar to compute-cloud-based VMs, the BigData machines are used to run experimental, ad hoc analyses on demand. Although QUT's HPC facilities provide magnitudes more processing power, they also require additional structure and enforce extensive restrictions that often conflict with the development of an in-progress algorithm or research

exploration. The BigData machines have been used to produce over 8TB of analysis results. When an analysis becomes stable and the scale of the data that is produced is increased, QUT's HPC compute resources are preferable.

C. Analysis

We have several forms of automated analysis categorized into two large groups: event detection and acoustic index generation. Event detectors produce time and frequency bounding boxes around spectral components of interest in an audio signal. Event detectors have been developed for a number of species: koalas (male), frogs, cane toads, cicadas, ground parrots, crows, kiwis, Lewin's rails, as well as generalized event detectors like Acoustic Event Detection (AED) and Ridge Detection [1, 23]. Acoustic indices, in contrast to detecting faunal events in audio streams directly, instead calculate summary statistics from the audio stream to provide large-scale insight into normally opaque audio.

Almost all analyses we produce are programmed in either C# or F#. C# is an unusual choice for research programming. However, contrary to the stigma of being too expensive, significant amounts of the C# and .NET toolchain have become free in recent years. C# has reasonable speed profiles, good tooling support, includes static analysis, and has automated garbage collection. It has a C-like syntax which is beneficial to researchers with a background in C or C++. The advent of multi-operating system support through the Mono project (http://www.mono-project.com/) has allowed our analyses to run on Unix/Linux operating systems. Where the performance of C# does not match that of native libraries (e.g. those written in C or C++), for critical operations our codebase will call native versions of the required functionality. For example, Fast Furrier Transforms (FFTs) are calculated by a native library for all of our analyses. Optimizations are implemented only when necessary as indicated by profiling.

The R language for statistical computing is used for the initial exploration of datasets. We have run large-scale data analysis in R; however, after the initial research stage has ended, often the research artifact transcoded to C# for ease of maintenance and extension by our researchers. Intensive or complex audio work is delegated to specialized programs, such as *SoX*, *FFmpeg*, *mp3splt*, and *shntool*. These programs are cross platform, provide a scriptable command line interface, and operate on files. We have wrapped these tools in two dedicated APIs – one for .NET and one for Ruby programs. Our Ruby audio-tools wrapper is open source (https://github.com/QutBioacoustics/baw-audio-tools).

Reproducibility of experiments and provenance of data are encoded in the tools and processes we use. Source audio data is considered immutable, with provenance maintained through log files and database metadata. Each compilation of the analysis programs includes the Git (a distributed source control application) commit hash. This provides a direct link from results and log files back to the source code that was used. All configuration files, output from analysis, and log files for each analysis are saved permanently. Most analyses return summary data (approximately 64MB per 24 hours of audio) however some return much more data (for example, the analysis approach presented by Dong [23] generates 6GB per 24 hours of audio).

In the spirit of avoiding premature optimization [24], very little optimization is implemented initially. As algorithms become stable, performance concerns may appear through analysis of larger datasets. The optimizations to apply are chosen through profiling and greatest return for time spent. Two examples of optimizations that adhere to this principle have significantly enhanced our analysis ability: 1) segmenting of input audio files and 2) parallelization.

Long input audio files require significant amounts of RAM to processes as one block; it is not feasible to analyze input audio longer than 2 hours in duration as one block. Additionally, ecological project requirements place increasing emphasis on large-scale continuous recording – often producing files 24hrs in length. To solve this problem all analyses have been standardized on processing one-minute blocks of audio. Thus, an analysis of a 24-hour file consists of 1440 smaller one-minute analyses. Specialized programs such as *mp3splt* discussed earlier avoid sequential seeking by using indexing to allow efficient cutting of arbitrarily large audio files. The result of this optimization is effectively large scale '*streaming*' of the input audio.

A substantial side effect of segmenting input audio is that each one-minute file can be analyzed independently. A master task is responsible for creating a list of work items. Each work item cuts the audio, runs the appropriate analysis, and returns results. The master task iterates through the work items and aggregates the results. This clean separation of concerns makes it exceptionally simple to parallelize analyses and fully consume all available resources. This *intra-parallelization* dedicates one thread per logical CPU to run analysis tasks concurrently.

Although intra-parallelization sufficiently consumes the resources of most average machines, it does not fully utilize the available resources on the BigData machines. Here the ad hoc

TABLE I. SPECTRAL INDICES ANALYSIS PERFORMANCE WITH VARYING PARALLELIZATION TECHNIQUES

Machine	CPU	RAM	I/O	Analysis		Time taken[a] (m/24h)	Effective Speed up
				Threads	*Instances*		
Normal Workstation	- i5-M560 - 4 logical processors - @ 2.67Ghz each	4GB DDR3	- Hitachi HTS545025B9A300	1	1	75.05	1.00×
				8	1	41.33	1.82×
				8	>1	N/A - Unreasonable demand	
BigData	- E5-665 - 32 logical processors - @ 2.4Ghz each	256GB DDR3	- 1Gbps Ethernet - 16GB RAM cache - No local disk	1	1	74.47	1.01×
				32	1	11.61	6.46×
				32	5	3.14[b]	24.00×

a) Minutes of analysis time needed to process 24 hours of audio
b) Experiement consisted of 20 files, each 24 hours, processed in batches of 5. Total time = 62.75 minutes. 62.75 minutes ÷ 20 files = 3.14 minutes/file.

scripts that already run analyses across thousands of files (1 day of audio per file) per job were parallelized. This *inter-parallelization* runs multiple instances of the analysis process on different files. Through tuning, it was determined that each BigData machine can process five instances of an analysis executable concurrently; that is, five inter-parallelized processes, each of which has intra-parallelization enabled as well. Tuning reveals that for the BigData machines the limiting resources is CPU. The relative speed gains from inter and intra parallelization are summarized in Table 1.

D. Visualization

Visualizing acoustic data is an effective way to see details and to obtain an overview of larger datasets. Even small amounts of data are considered opaque and hard to reason about without analysis [10, 25]. Datasets that are months, even years long are common and produce numerical data that is incomprehensible. For large datasets, visualizations are increasingly becoming the only way to interpret results.

We calculate *acoustic indices* for one-minute blocks that represent content of ecological interest. Each acoustic index summarizes an aspect of the acoustic energy distribution in audio data. Three acoustic indices can be represented by different color channels. Presenting the combination of indices over time as colors in an image can expose the content of the audio and allow for navigation of audio that can be years in duration [9]. Indices can be calculated from the spectral content or waveform; there are a range of methods for calculating indices in the literature. Typical measures include SNR and amplitude. The dispersal of acoustic energy in a recording – the temporal entropy – is a promising candidate [26], as it has a good correlation with avian activity.

The choice of which three indices to combine requires measures that can be compared. We chose three indices which can easily be normalized to the range $[0, 1]$: temporal entropy, *spectral entropy* (H[s]) (a measure of acoustic energy dispersal through the spectrum) [26], and the *acoustic complexity index* (ACI), which is a measure of the average absolute fractional change in signal amplitude from one frame to the next through a recording [27]. These False-color spectrograms (see Fig 4) are built from more than one measure of the acoustic content, whereas pseudo-color spectrograms are mappings of the spectral power values to color. The combination of three indices will provide more information than a pseudo-color spectrogram if the indices used are independent.

An advantage of false-color images is that they tolerate and can even highlight data corruption and missing data. It is common to manually remove noisy or clipped recordings containing excess mechanical noise, wind, and rain, however this does not scale.

V. Work In Progress

A. Current Website Architecture

A core goal of our ecoacoustics research is to make accessing, visualizing, and analyzing large-scale acoustic data accessible to scientists. To do this we use the QCIF cloud infrastructure to host our publically accessible website. This open source application, the *bioacoustic workbench* (https://github.com/QutBioacoustics/baw-server), is designed to provide access to large-scale ecoacoustic datasets. The website successfully allows random-access to any of the ingested audio data – currently 15TB of audio.

The website provides tooling for creating *projects* and *sites* to manage audio data. From a site, access to any audio recording is possible: when loaded a visual depiction accompanies the playback of audio. Audio can be played indefinitely for radio-like listening, or can be played in sections to allow manual analysis of a segment. Annotations can be drawn on the spectrogram that, when tagged with a species name, can identify a faunal vocalization. The annotation process is useful for generating training datasets used by automated analyses [23].

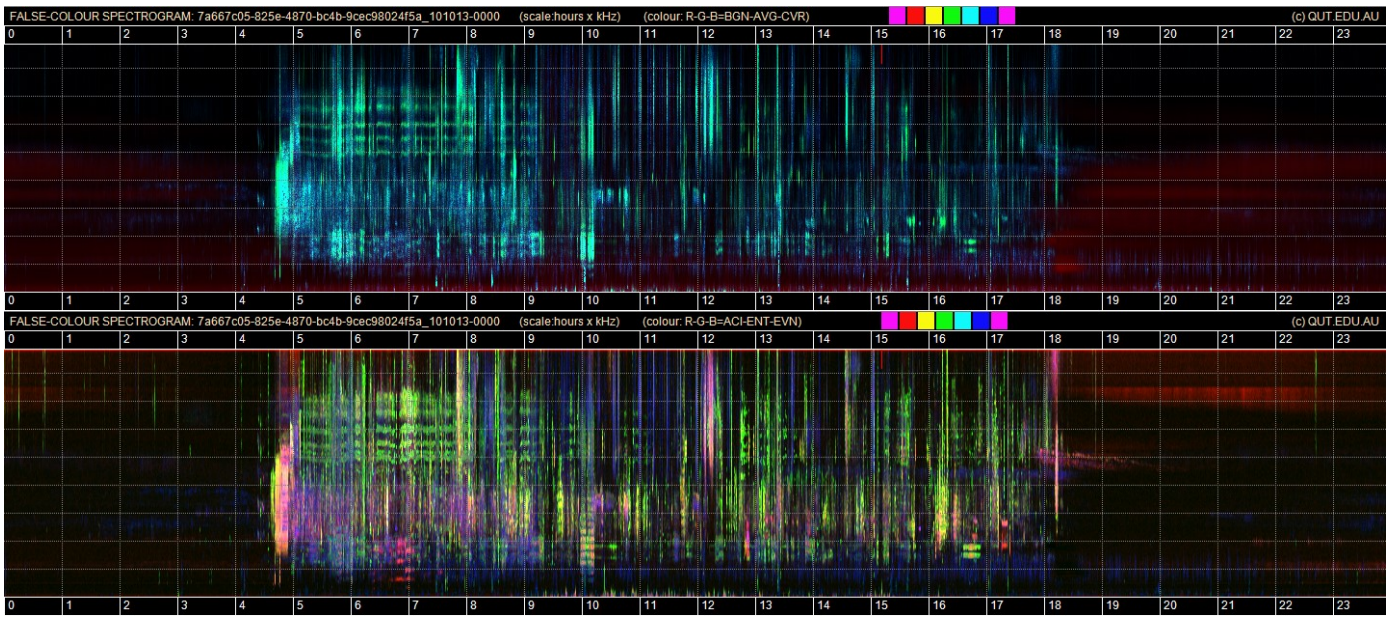

Fig. 4. Two false-color long duration spectrogram. These spectrograms use spectral indexes to visualise acoustic activity over a 24 hour period

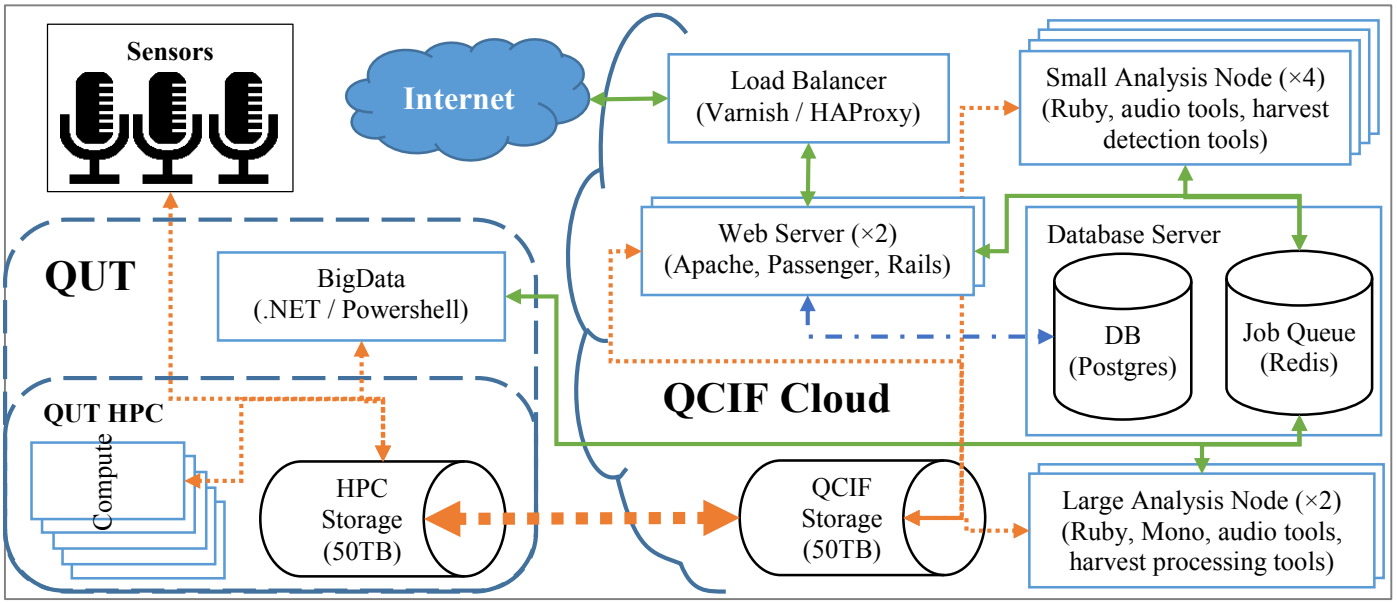

Fig. 5. Diagram of cloud scale architecture. Orange (dashed) lines represent acoustic data, green (solid) represent metadata, blue (dash-dot) represent database access

TABLE II. PLANNED ARCHITECTURE FOR SCALABLE ECOACOUSTICS WEBSITE HOSTED IN THE QCIF CLOUD

Location	VM Flavor	Instances	Resources (per instance)	Resque Queues highest priority first: `QUEUE_NAME`×Concurrency	Est. Time per Request/Job
QCIF	Web Server	2	2 VCPUs, 8GB RAM	N/A – these servers will create job items	< 2s
	Database	1		N/A – Resque host	< 1s
	Small Analysis Node	3	1 VCPU, 4GB RAM	`MEDIA`×1, `HARVEST_WATCHERS`×1	6-18s
	Large Analysis Node	2	4 VCPUs, 16GB RAM	`MEDIA`×4, `HARVEST_FILE`×4, `ANALYSIS_JOBS`×1, `MAINTENANCE`×4	1-20m
QUT	BigData Machines	1 exclusive 11 shared	32 CPUs, 256GB RAM	`ANALYSIS_JOBS`×5, `MAINTENANCE`×4	1-20m

The website is built using the Ruby on Rails framework. It utilizes our audio-tools API to cut and cache media. This provides responsive playback and on-demand loading of previously unseen segments of audio. Currently the webserver controls and executes the cutting of audio and generation of spectrograms. This is inefficient and will be extracted to separate, dedicated servers in the future.

B. *Future Architecture*

Our project has recently migrated to the QCIF cloud. The bioacoustic workbench and all audio data are currently hosted on QCIF resources; however, we have yet to fully utilize the resources available. Increased user demand and I/O strain on webservers has necessitated continued scaling. In practice, much of the analysis is driven by internal research needs and consequently run within QUT on BigData or HPC resources.

However, recent publications and increased interest in our work has resulted in progress towards more formal, scalable infrastructure. Additional functionality, including the ability to run analyses and generate false-color spectrogram images, will improve the navigation and utility of the public website.

Analysis will continue to be done locally to make use of the flexibility BigData machines afford, following the hybrid approach. We still have the need for ad-hoc scripts; however, exposing concrete analyses will improve the utility of the Bioacoustic Workbench for all users.

The job running system under development is built on Resque (https://github.com/resque/resque), a Ruby library. It uses priority queues (backed by a Redis in-memory database) to handle various asynchronous tasks. Analysis programs, audio cutting, spectrogram generation, harvesting, and maintenance jobs will be enqueued with Resque. Dedicated analysis VMs will be provisioned in the QCIF cloud to process jobs. The server architecture is shown in Fig 5 and the planned VM provisioning table and job queue distribution is shown in Table II. Additionally, Resque job runners will be installed on BigData machines to ensure compute power is never wasted – thus creating a hybrid *cloud and local* job system.

VI. CONCLUSION

Production systems for research work are difficult to provision and maintain due to the constantly changing nature of active research. The capture, analysis, and use of results from big data activities is widespread; however, practical descriptions of on-going research by groups with complex applications are needed. This paper has given an overview of the Ecoacoustic Research Group's approach to big data analysis.

The management of raw audio data, analysis programs, methods of executing programs in parallel, and resulting output is an important, significant, and time-consuming part of analyzing large data sets. It requires knowledge and experience from a range of domains implemented by a range of professionals.

Compute resources are available from a number of organizations and can provide the basis for effective big data processing. The disparate resources are often required to inter-operate. Few researchers have the background to be able to manage compute, storage, and cloud resources. As the amount of data used in the majority of disciplines increases, professional support for researchers also needs to increase.

Visualizations are an effective way to reveal patterns and summarize data that is otherwise opaque and difficult to interrogate. Developing methods for generating useful visualizations is critical to evaluating analysis algorithms. Increasing pressure to provide results from analysis of large datasets can spur researchers to remain within constraints set by professional staff; however, research requires a constant develop-and-test cycle. This tension can be addressed through freezing features and refactoring checkpoints.

ACKNOWLEDGMENTS

The authors wish to acknowledge the dedication and hard work of all members in our research group (http://www.ecosounds.org/people/people.html). In particular, we thank Jason Wimmer for conducting fieldwork and our citizen science collaborators; these are birders, conservation groups, and individuals that help analyze and verify data produced by analyses.

The authors gratefully acknowledge the funding and resources provided by Queensland Cyber Infrastructure Foundation (QCIF) and the National eResearch Collaboration Tools and Resources (NeCTAR). Grant: *QCIF NeCTAR Tools Migration Project "Acoustic WorkBench (AWB)"*.

The authors also acknowledge the resources provided by the Big Data Lab at the School of Electrical Engineering and Computer Science, QUT. Additionally we acknowledge the support and resources provided by QUT's High Performance Computing group.

REFERENCES

[1] J. Wimmer, M. Towsey, B. Planitz, I. Williamson, and P. Roe, "Analysing environmental acoustic data through collaboration and automation," *Future Generation Computer Systems,* vol. 29, pp. 560-568, 2// 2013.

[2] R. Butler, M. Servilla, S. Gage, J. Basney, V. Welch, B. Baker*, et al.*, "Cyberinfrastructure for the analysis of ecological acoustic sensor data: a use case study in grid deployment," *Cluster Computing,* vol. 10, pp. 301-310, 2007.

[3] Wildlife Acoustics. (2011, 23/05/2011). *Song Scope Product Page*. Available: http://www.wildlifeacoustics.com/songscope.php

[4] A. Gasc, J. Sueur, S. Pavoine, R. Pellens, and P. Grandcolas, "Biodiversity Sampling Using a Global Acoustic Approach: Contrasting Sites with Microendemics in New Caledonia," *PLoS ONE,* vol. 8, p. e65311, 2013.

[5] M. Towsey, S. Parsons, and J. Sueur, "Ecology and acoustics at a large scale," *Ecological Informatics,* 2014.

[6] D. Tucker, S. Gage, I. Williamson, and S. Fuller, "Linking ecological condition and the soundscape in fragmented Australian forests," *Landscape Ecology,* vol. 29, pp. 745-758, 2014/04/01 2014.

[7] I. Potamitis, S. Ntalampiras, O. Jahn, and K. Riede, "Automatic bird sound detection in long real-field recordings: Applications and tools," *Applied Acoustics,* vol. 80, pp. 1-9, 6// 2014.

[8] M. Cottman-Fields, A. Truskinger, J. Wimmer, and P. Roe, "The Adaptive Collection and Analysis of Distributed Multimedia Sensor Data," in *E-Science (e-Science), 2011 IEEE 7th International Conference on*, 2011, pp. 218-223.

[9] M. Towsey, L. Zhang, M. Cottman-Fields, J. Wimmer, J. Zhang, and P. Roe, "Visualization of Long-duration Acoustic Recordings of the Environment," *Procedia Computer Science,* vol. 29, pp. 703-712, // 2014.

[10] J. Foote, "An overview of audio information retrieval," *Multimedia Systems,* vol. 7, pp. 2-10, 1999.

[11] T. Hey. (2014, 22/07/2014). *Beyond Open Access to Open Data*. Available: http://hdl.handle.net/2142/47423

[12] E. Dumbill. (2012, 22/07/14). *What is big data? An introduction to the big data landscape*. Available: http://radar.oreilly.com/2012/01/what-is-big-data.html

[13] Y. Demchenko, P. Grosso, C. de Laat, and P. Membrey, "Addressing big data issues in scientific data infrastructure," in *Collaboration Technologies and Systems (CTS), 2013 International Conference on*, 2013, pp. 48-55.

[14] S. Kelling, W. M. Hochachka, D. Fink, M. Riedewald, R. Caruana, G. Ballard*, et al.*, "Data-intensive science: a new paradigm for biodiversity studies," *BioScience,* vol. 59, pp. 613-620, 2009.

[15] G. Bell, J. Gray, and A. Szalay, "Petascale computational systems," *Computer,* vol. 39, pp. 110-112, 2006.

[16] Xeno-canto Foundation. (2014, 22/07/14). *Colophon and Credits*. Available: http://www.xeno-canto.org/about/credits

[17] S. Duan, J. Zhang, P. Roe, J. Wimmer, X. Dong, A. Truskinger*, et al.*, "Timed Probabilistic Automaton: A Bridge between Raven and Song Scope for Automatic Species Recognition," in *Twenty-Fifth IAAI Conference*, 2013.

[18] L. J. Villanueva-Rivera and B. C. Pijanowski, "Pumilio: A Web-Based Management System for Ecological Recordings," *Bulletin of the Ecological Society of America,* vol. 93, pp. 71-81, 2012/01/01 2012.

[19] L. Dayton, "Giant telescope to 'create hundreds of jobs'," in *The Australian*, ed, 2012.

[20] P. J. Guo and D. R. Engler, "Towards Practical Incremental Recomputation for Scientists: An Implementation for the Python Language," in *TaPP*, 2010.

[21] S. R. Kohn, G. Kumfert, J. F. Painter, and C. J. Ribbens, "Divorcing Language Dependencies from a Scientific Software Library," in *PPSC*, 2001.

[22] T. Hey, S. Tansley, and K. M. Tolle, "Jim Gray on eScience: a transformed scientific method," in *The Fourth Paradigm: Data-Intensive Scientific Discovery*, ed Redmond, Washington: Microsoft Corporation, 2009.

[23] X. Dong, M. Towsey, Z. Jinglan, J. Banks, and P. Roe, "A Novel Representation of Bioacoustic Events for Content-Based Search in Field Audio Data," in *Digital Image Computing: Techniques and Applications (DICTA), 2013 International Conference on*, 2013, pp. 1-6.

[24] D. E. Knuth, "Structured Programming with go to Statements," *ACM Computing Surveys (CSUR),* vol. 6, pp. 261-301, 1974.

[25] M. Towsey, J. Wimmer, I. Williamson, and P. Roe, "The use of acoustic indices to determine avian species richness in audio-recordings of the environment," *Ecological Informatics,* vol. 21, pp. 110-119, 5// 2014.

[26] J. Sueur, S. Pavoine, O. Hamerlynck, and S. Duvail, "Rapid Acoustic Survey for Biodiversity Appraisal," *PLoS ONE,* vol. 3, p. e4065, 2008.

[27] N. Pieretti, A. Farina, and D. Morri, "A new methodology to infer the singing activity of an avian community: The Acoustic Complexity Index (ACI)." *Ecological Indices,* vol. 11, pp. 868–873, 2011.