Direct, Distributed Access to Analogue Signal Processing Hardware through the Cloud

Sean Peuquet[1], David Jones[2]

[1] Ludic Sound, Denver, CO, United States
seanpeuquet@ludicsound.com
[2] Mad Wire, LLC, Fort Collins, CO, United States
david@drj.io

Abstract

This paper discusses the development and use of the RackFX software platform, which provides digital access to real analogue hardware devices through a web interface that controls remote robotic components to handle audio processing requests. The platform is presented as an alternative to digital plugins that emulate analogue hardware devices (analogue modelling) as a way to incorporate analogue sound characteristics into computer-based music production. Given existing technological, social, and perceptual tensions between DAW-based effects plugins and outboard analogue processing units, the RackFX technology enables more widespread psychoacoustic comparison between analogue and digital signal processing by extending access to analogue gear. A technical overview of the platform is provided, which outlines user experience and various server and on- site robotic processes that ultimately support the return of an analogue-effected digital audio file to the user for each processing request. Finally, the social implications of using the platform are discussed, regarding the cultivation of a community of users both sharing and accessing analogue hardware with unprecedented ease. RackFX, ultimately, represents a new way to leverage digital technology toward the democratization of music production tools and techniques.

Keywords: cloud computing, analogue signal processing, Arduino, web application, robotics.

Introduction

The fetishization of analogue audio recording, production, and reproduction technology shows no sign of abating. In the large and ever expanding field of music technology, analogue hardware continues to be associated with musically desirable psychoacoustic descriptors, most notably 'warmth'. While the term warmth is strongly correlated to the acoustic phenomena of harmonic distortion and high frequency roll-off, the idiosyncratic production and (inter)subjective perception of analogue warmth poses interesting problems for the computer musician.

Digital music technology replicates, processes, and stores audio exactly according to its software programming and the hardware limitations of the computer the code runs on. Which is to say, digital music is (barring any hardware stability issues) deterministic–from the moment directly after analogue to digital conversion until the moment the signal is converted back to analogue for sound reinforcement. Modelling the effect of psychoacoustic warmth using digital signal processing (DSP) techniques thus poses a hierarchical problem regarding the accurate representation of sound; no longer is the mere capture and digital representation of an analogue signal at issue, but rather the problem concerns the capture and representation of how that analogue signal was produced, which necessarily entails some degree of indeterminacy. As Karjalainen and Pakarinen describe: "… virtual analogue modelling seems straightforward but is found demanding due to the nonlinearities and parametric variation in the analogue domain" (Karjalainen and Pakarinen, 2006: 153). The desired perceptual excess of an analogue processed signal, it's warmth, is largely a direct result of the physical components of the analogue system, their unpredictability and imperfection. In this respect, the modelling of analogue effects (warmth correlates) using DSP is also closely related to synthesis, specifically physical modelling synthesis.

While the modelling approach has led to great successes and a burgeoning marketplace for software instruments and analogue modelled plugins alike (see Waves, 2016; UAD, 2016), there remains both a precision problem and a perception problem regarding the refinement and accuracy of our models. The question remains: what physical interactions are necessary to model and to what degree of accuracy—sufficient to overcome the just noticeable difference (JND) in respect to some analogue reference point? Despite Julius O. Smith's 1996 pronouncement (regarding synthesis) that, "we appear to be approaching parity between real and virtual acoustic instruments," (Smith, 1996: 44) we are twenty years on and it appears that the lack of parity is increasingly what structures both the popular discourse and commercial reality of music recording and production. The cello has yet to be fully replaced; in the same way that people who actually have access to a vintage Fairchild 670 would claim that all attempts to emulate the device as a digital plugin have failed. So despite the ease and accessibility of plugin emulators, actual vintage analogue hardware processing units remain the gold- standard.

Counter to the prevailing trend of digitally modelling analogue processes that yield the sensuous qualities of sonic warmth, the authors have sought to simply digitize access to the analogue components and processing itself. The RackFX platform is essentially a "straight from the horse's mouth approach" to analogue signal processing. While the idea of enabling distributed access to physical acoustic resources is not without precedent (see Silophone, n.d.) or the the MIT Responsive Environments Group's "Patchwerk" web interface to Joe Paradiso's massive modular synth (Patchwerk, n.d.), the RackFX platform is a uniquely scalable and flexible solution with potentially longer-term consequences and implications. The technology was conceived of by David Jones, and developed by Jones and Sean Peuquet across much of 2015 and 2016. The platform continues to be developed with new features being added on a regular basis, and is currently accepting Beta Registrations with a scheduled Beta release in late November of 2016. Across the rest of this paper, the RackFX platform will be presented as both a technological solution and a paradigm shift regarding issues of access, affordability, and quality that govern the viability of signal processing using analogue hardware devices.

Technical Overview

The RackFX platform allows users to log-in to a web application, upload audio to the cloud, and process that audio using real analogue hardware located, potentially, anywhere in the world. On the server-side for any RackFX processing job, the web-app proxy receives all internet requests and forwards them into a web-app cluster written in node.js. The web-app cluster interfaces with Amazon S3 for storage and a MySQL database handling all site data. The queue software, a lightweight file-based JavaScript Object Notation (JSON) queuing service, runs on the web-app cluster. When a job is submitted for processing, the queue schedules it, waits for the targeted analogue device to become available and then passes each job to the appropriate device (computer hosting the device, potentially anywhere in the world) to be processed one at a time. The queue passes the job specifics to a Messaging Application Programming Interface (MAPI) cluster (also written in node.js), which messages the appropriate Machine Device Controller (MDC) included as part of the "RackFX Studio" software application running on an OSX machine in close physical proximity to the actual analogue hardware units.

At this point, the processing request has moved from the server (web-app) to the actual device-side of the processing system. It is important to note that the device-side processing can be physically located anywhere (provided there is an internet connection) and the web-app is capable of routing processing requests to multiple device-hosting locations. One a device-side machine receives a processing request, the MDC (also written in node.js) orchestrates the actual processing of the (still) digital audio in the following order: 1) Identify the device selected for processing and queue the job until it becomes available; 2) Download the referenced digital audio file from the cloud; 3) Turn on electrical power (using a Web Power 7 switch) to the given analogue hardware unit and an Arduino interfacing with the device using device-specific robotic components; 4) Pass device parameters to the Arduino (using Johnny-Five, a JavaScript robotics module for node.js); 5) Wait for the robotics to physically interact with the analogue hardware device and set all parameters; 6) Spawn a Cycling74 Max7 standalone application (that we call "audio-io") that handles the real-time digital audio playback and capture of the analogue-effected signal.

Once the roundtrip signal i/o is complete, audio-io tells the MDC that the new file has been written, the MDC cleans up: uploads the new file to the server, signals completion, and shuts all on-site devices down. A visual overview of the whole RackFX system, including major components and signal flow, is shown in Figure 1.

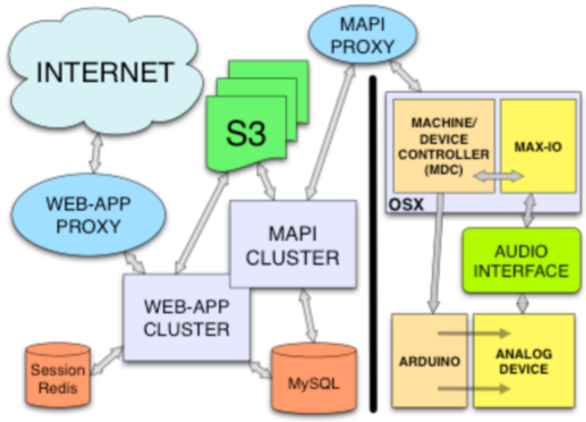

Figure 1. RackFX server-side and device-side system overview.

Robotic Device Interface Specifics

In order to maximize automation of the interaction between the user (client-side) and the analogue device (device-side), RackFX aims to outfit analogue devices with custom robotics that physically interact with the device's particular control panel. Various models of stepper motors, actuators, and sensors combine to create each hardware interface between machine and device. These robotic components are controlled using a dedicated Arduino board for each device (see Figure 2).

Figure 2. Arduino and stepper motors configured for control of a Fender Princeton Reverb amplifier.

Each arduino is loaded with the Advanced Firmata firmware (a protocol for communicating with microcontrollers from software on a host computer) and addressed by its host computer using the JavaScript client library Johnny-Five (J5), a robotics and Internet of Things programming framework. The node.js MDC loads the J5 module to enable communication between the MDC and each device. When the MDC goes to process a job (the next job in the queue), the program identifies the Arduino associated with the specified analogue device, instructs all stepper motors to reset (one at a time) by (over-)turning all knobs counter-clockwise a full rotation to ensure the analogue device potentiometers are set to zero. The MDC then instructs each stepper to turn a certain number of steps commensurate to the parameter setting specified by the user through the web-app. (Maximum and minimum step values are tuned in relation to each physical parameter setting for each device, as part of the configuration process.) After a short delay to ensure all parameters are set, the MDC communicates with audio-io to commence audio processing.

Discussion

The RackFX platform allows users to access analogue equipment through an easy to use web site. This platform allows users to use audio processing equipment through cloud- based technology and robotics. And in the future, studios and individuals can bring their devices to the community and become a RackFX partner, bringing analogue processing capability to users around the world through our easy-to-use custom framework.

Ultimately, RackFX represents an opportunity for musicians and audio producers to engage in “in-the-box” analogue signal processing through the web. In the past, low-budget musicians, video producers, music producers and podcasters had to rely on increasingly expensive digital plugins that attempt to emulate analogue signal processing devices, or they had to invest in cheap analogue gear with low-quality components in an attempt to achieve the sound qualities they associate with high-budget studio analogue gear. Now users can have access to this high-end equipment through the RackFX platform.

As a music production solution, the RackFX project seeks to leverage digital scalability and distributed access toward great analogue accessibility. While commitment to achieving ever more refined in-the-box DSP techniques and analogue device emulations will (and should) continue, we should not think that *parity* between the digital and analogue world is either necessary or desirable. Nor should we eschew what digital tools have afforded in the name of maintaining limited access to analogue processing units– resulting in analogue fetishization to an even greater degree, given such a scarce resource. By leveraging a host of digital technologies, including cloud computing, real-time digital audio manipulation, and robotics, the RackFX platform provides an alternative path: make analogue devices accessible through the web to empower all musicians, regardless of budget. At the very least, our psychoacoustic value judgements regarding the ‘warmth’ and ‘presence’ of analogue processing effects will be put to the test now that analogue gear is no longer cloistered. Ideally, a platform like RackFX will help advance our ability to hear.

Bibliography

J.O.Smith, III. “Physical Modelling Synthesis Update.” *Computer Music Journal*, vol. 20, no. 2, pp. 44–56, 1996.

M. Karjalainen and J. Pakarinen. “Wave Digital Simulation of a Vacuum-Tube Amplifier,” *ICASSP*, vol. 2, pp. 153–156, 2006.

“Patchwerk: Control a Massive Modular Synthesizer.” Retrieved from http://synth.media.mit.edu/patchwerk/ on April 4, 2016.

“Silophone project.” Retrieved from http://www.silophone.net/eng/about/desc.html on April 20, 2016.

Universal Audio. “Analogue Classics Bundle.” Retrieved from http://www.uaudio.com/uad-plugins/plug-in-bundles/analog-classics-pro-bundle.html on November 2, 2016.

Waves. “Waves Audio Analogue Models Plugins.” Retrieved from http://www.waves.com/plugins/analog-models/ on February 29, 2016.