An Online Architecture for Remote and Ubiquitous Analog Audio Mixing

Andrei A. K. Yefinczuk[1], Marcelo O. Johann[1], Rodolfo P. Pirotti[2], Lucas F. Zawacki

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
[1]Post-Graduate Program on Microelectronics (PGMICRO)
[2]Post-Graduate Program in Computer Science (PPGC)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

andrei.yefinczuk@gmail.com, johann@inf.ufrgs.br, rodolfo.p.pirotti@gmail.com, lfzawacki@gmail.com

***Abstract**. This paper presents an architecture and a running prototype of an online system which provides access to analog audio mixing hardware. The proposed architecture optimizes equipment usage and maximizes available mixing hours, and so increases the cost benefit when compared to a traditional mixing studio. The users send the audio files and parameters to a server which, connected to a hi-fi analog mixing system and high quality audio interfaces, runs the mixing process and allows access to the output files. An architecture like this can democratize access to the process of analog mixing and allow musicians to try it themselves with their own audio files but using remote hardware. The current prototype and initial impressions will be presented herein as well as opportunities for ubiquitous music production by making use of this online architecture.*

1. Introduction

The music production process has been changing a lot over the last decades. The introduction of digital equipments and the decreasing of costs allowed people who did not have access to recording equipment in the past to start trying music production on their own by making instrument recording and audio mixing at home, being able to even produce entire music albums [Valladares, 2011]. Not exclusively to home users, tools like Digital Audio Workstations (from now on called DAW) and Virtual Studio Technology plug-ins (VST), both integrated with computers and audio interfaces, are nowadays part of a professional music studio workflow for music recording and production.

Although this new kind of digital tools brought several possibilities and increased the popularity of music production, generally real instruments (including acoustic instruments, effect processors, compressors and mixers) has a better sound quality and smoother control. But they also have higher prices and are often difficult to find and access [Zawacki and Johann, 2014].

In [Zawacki and Johann, 2014], a system for analog audio recording using remote servers is proposed. The proposed system enables users to have access to high quality audio equipment by using remote servers on a batched access model. This access model optimizes equipment usage by requiring the user to submit a job request and retrieve the expected result later. This is different from a remote real-time access, which requires timing and latency considerations, already addressed in works such as [Kapur, Wang, Davidson, and Cook, 2005] and [Orto and Karapetkov, 2011]. The batched access increase system usage, by letting the user perform the first steps of the music production (music composition, individual audio track recording, parameter setting) in his own equipment (or computer). Later, a job request is submitted to the remote system, which will process and execute the mixing job when available, and notify the user to download the output file.

This modular and distributed architecture allows musicians to access high quality analog mixing equipments, effect processors and any other high quality equipment used on the final mixing activity of music production. It opens the possibility for users to perform their own mixing using high quality audio equipment, which usually is available only on professional and expensive studio environments. Besides, anyone who owns high quality equipment, even home made or customized ones [Johann, Yefinczuk, Pirotti and Chiaramonte, 2015], would be able to grant access to remote musicians by running a remote mixing system according to the architecture proposed herein.

In addition to that, it is possible to use the same input files (e.g., individual track recording) and parameter setting to be processed and tested on different equipments, performing an equipment comparison to choose the one which better fits the needs. Also, it is possible to request mixing jobs and listen to the results in any online connected device, such as computers, tablets and smartphones. Once the job is executed on the remote server, users are able to access the output anywhere and listen to the expected result. If it is not as expected, it is possible to request a new job to be executed and select different equipments and parameters.

1.1 Motivation

This work is motivated by the idea that we can expand the notion of Ubiquitous Music to include remote access to high quality audio systems. This is an ongoing effort marked by our previous works in [Zawacki and Johann, 2014]. Typically ubiquitous music is referred to and implemented in the context of mobile computing and involves sound generation in place by the user's device or via a sound installation. Often the aural quality of such systems is not a priority and is limited by processing, storage and audio hardware constraints of mobile computing devices.

What we are proposing here is a system for music composition that trades immediate sound feedback for sound quality and opens the possibility of mixing the user's own music from any device with internet connectivity. We believe this could enhance the concept of “ubiquitous music”, because it enables musicians and producers to engage with and access traditional production tools with the added possibilities of modern software.

Furthermore, an always online and remote high quality mixing system can be viewed in the light of the interaction metaphors devised in [Keller, Lazzarini and Pimenta, 2014] as it deals primarily with sequencing, time tagging and mixing of musical events, the challenge being on how to make it fit a musicians workflow enough that it could be said as invisible and transparent [Costa, Yamin and Geyer, 2008], or if that is even possible.

2. Audio Mixing

A mix is a sonic presentation of emotions, creative ideas and performance [Izhaki, 2008]. A more technical definition of mixing is a process in which multitrack material, whether recorded, sampled or synthesized, is balanced, treated and combined into a multichannel format, most commonly two-channel stereo.

It is difficult to know when and where the art of mixing started. We can look at the instrumentation of orchestral pieces as a basic form of mixing. The placement of musicians in a room reinforces this idea, considering the levels and depth of instruments in the final recording [Izhaki, 2008]. In a musical production, it is in the mixing stage that these effects of volume, depth and position of the recorded sounds are performed. This is a critical task both technically and artistically. Mixing consoles, compressors, equalizers, reverbs, effects, etc. are tools used for this purpose.

The sum operation of multiple signals, which is the focus of this paper, is one of the key points of the technical process of mixing and can be made both in the analog domain, when an electronic device does the operations and the final sum of the signals, as well as in the digital domain, when the process is done through mathematical calculations on the computer. A mixed approach is often used nowadays. This paper proposes an architecture that enables the user to make his own mix using analog equipments over the internet.

3. Remote Analog Mixing System Architecture

To achieve the goals mentioned before, we propose to let the artistic part of the mixing activity be done by the user, but allow it to be accessed from anywhere, anytime making use of the architecture shown in figure 1.

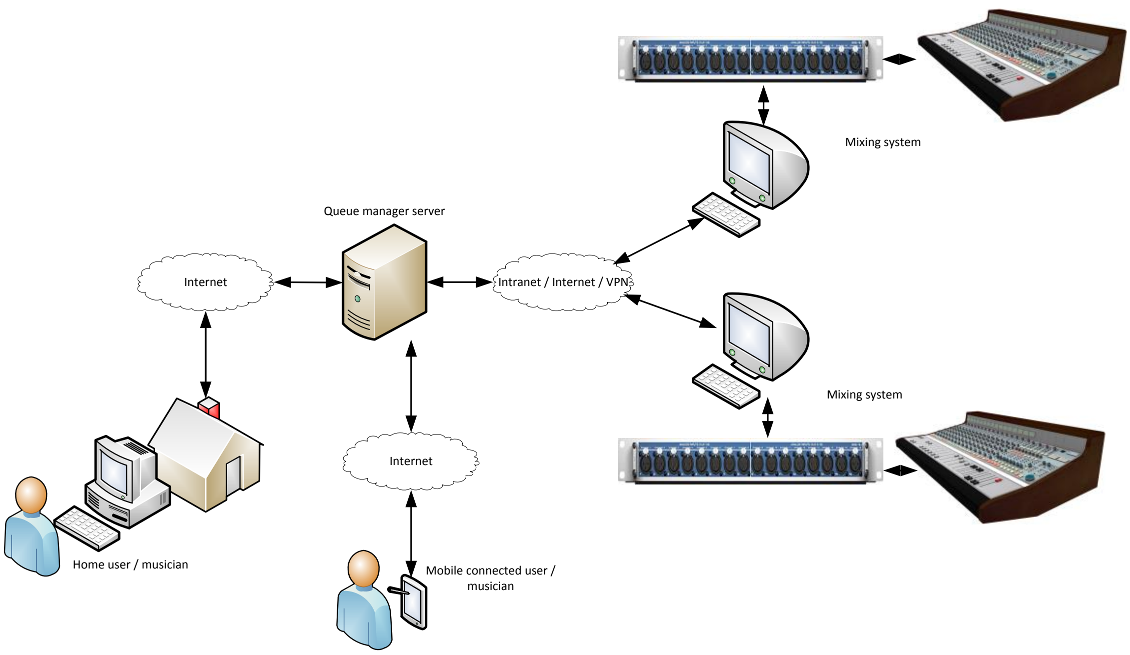

Figure 1 - Proposed system architecture

The proposed system uses a modular architecture, which allows it to increase its capacity and available mixing equipment models easily.

3.1. User mixing activity

As one of the most important parts of the mixing activity, the audio recording and parameter setting is still done by the user. This ensures that the artistic part of the mixing activity is experienced and controlled by the user, whereas our system only executes the sum of the analog signals, with high quality electronic equipments. It lets the user to choose the appropriate levels, pan settings and any other effect it wants to insert into the audio channels. Any setting can be tested and the result listened to, allowing this system to be used for educational purposes, equipment comparison and hi-fi audio mixing.

As a first step for the remote mixing, the user records all his audio tracks on individual audio files (one for each track to be mixed), including level and pan (panoramic) settings. After this is done, the user, by using a client interface, sends a request to the queue manager server for a job to be executed. On the job creation, all audio files are sent to the queue manager server. The client interface is capable of showing an estimated time in which the user's mixed audio will be available, based on the other requests present on the server queue. After the mixed audio is available, the user can log into the system to download the output file(s).

3.2. Queue managing server

The remote server is responsible for receiving and storing the audio files sent by the user. It can be connected to more than one analog mixing system, with different mixing equipment models. Based on the users' requests, the remote server sends the audio files to the appropriate mixing system at the appropriate time, considering other enqueued jobs and the bandwidth usage.

Once the mixing system is available, the mixing job starts. All the audio files are played together, according to users' requests. The audio data is sent to the analog mixing system using hi-fi audio interfaces, which is required to ensure the audio quality [Johann, 2010]. The audio channels are then mixed together, and the result is recorded back to the system, also using high quality audio interfaces. After the whole process is done, the output file is sent back to the remote server, which makes it available to the user to be downloaded.

4 Working prototype

A prototype was developed as a proof-of-concept. It shall help to identify the challenges, implementation options and usability issues later.

4.1 Audio recording

The individual digital track recording is done by the user on his own computer and to collect individual track data for the mixing activity, we have developed a VST plug-in. The plug-in needs to be inserted on the user project and the DAW configured (by the user) to send each audio track to the plug-in, by using the "send" slots. By doing this when playing the audio track, the plug-in is able to receive the audio data in the same way it should be mixed with the other tracks, considering parameters such as volume level and panoramic settings.

The plug-in then saves a separate audio file for each track, named trackN.wav (N is the track number). When all the audio tracks are recorded and processed by the plug-in, the user presses the "send" button on the plug-in user interface. The plug-in will then compress all the files (using standard zip method) and connect to the server to send the files. The file send was done by using CURL, an open source tool for file transfer.

4.2 Client interface

As our main goal is to test the proposed architecture, we implemented the client embedded on the VST plugin. Once the mixing activity is done and all the audio data recorded by the plugin, the user presses the "send" button on the plugin, which will connect to the server, send the audio data, and show to the user an URL which should be used later to get the output file. Figure 2 shows the user interface of the plug-in, with only two buttons: rec (to record the individual audio tracks) and send (send all audio data to the server).

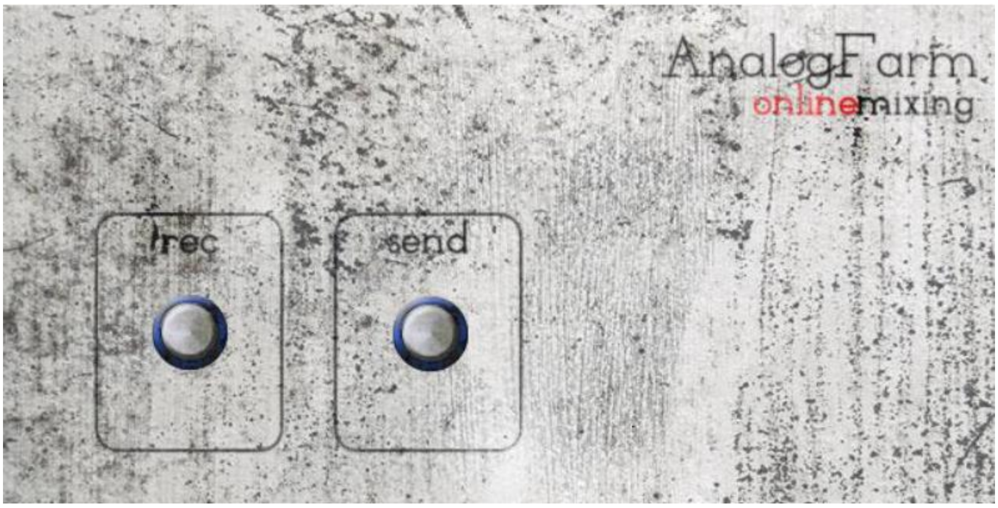

Figure 2 - Plug-in user interface

4.3 Server implementation

On the proposed architecture, the server which receives and manages the job queue is independent of the mixing system, to allow distributed mixing systems to be used. On the working prototype, we implemented the server and the mixing system on the same equipment.

The queue manager part of the server was implemented by using PHP and Java. The Java application receives socket messages from the PHP method to receive the job requests and audio data, enqueueing them to be processed when the mixing equipment is available. Once the mixing equipment is available, the java application runs a recording script, which is responsible for starting the mixing and recording activity.

The mixing and recording activity is done by using a script which automates an operation with the Cubase 6 Elements DAW. For this prototype, we choose to use a standard commercial DAW, so we do not need to worry about synchronizing the audio files externally and ensure latency compensation.

The script, created by using Applescript, is responsible for importing the user’s audio files into a new project on the DAW. An empty template project is used for that. Each audio track is addresses to a different output channel on the output audio interface, and an extra two tracks are used to record the mixing result, by using two input channels from the audio interface.

When the mixing process is done, a message is sent to the queue manager server, which makes the file available to the user.

4.4 Equipments used on the prototype

The implementation of the mixing system used a Behringer FCA1616, which has eight output channels, and one Behringer ADA8000, which has another eight output channels. These two interfaces were connected to a Mackie 1402 vlz analog mixer. The output audio from the Mackie was recorded by using two input channels of the Behringer FCA1616 interface.

4.5 Data transfer

The amount of data transfer needed for the implemented architecture must be considered. Since the mixing process is not done in real time, there is no need for high quality audio or video streaming services.

On a scenario in which one 24 track mixing system is used 50% of the time, around 4.07 TB of data would be transferred during one month. If only 8 tracks are used, the total data is reduced to 1.35 TB. Table 1 shows a comparison of how much data would be transferred in a month, considering the number of tracks and the system usage rate.

Table 1 – data transfer per equipment utilization

	8 tracks	16 tracks	24 tracks
5%	0.136 TB	0.271 TB	0.407 TB
10%	0.271 TB	0.543 TB	0.815 TB
25%	0.679 TB	1.358 TB	2.037 TB
50%	1.358 TB	2.716 TB	4.073 TB

On the user side, on a scenario where a 4 minute song is uploaded, with 24 tracks, on a 10 Mbits/s download bandwidth (around 128 Kbits/s upload), around 110 minutes are required to upload all the audio files. Considering the mixing activity as the final part of the music composition, this is considered an acceptable time even with current technology. With today’s cloud services, it could be possible to implement a server side way to access user data store in traditional cloud platforms, making it easier for the user to upload these files.

5. Analysis and discussion

The batched access model used on the proposed architecture improves the system usage by executing the requested job once and providing the result back to the user. To our knowledge, this is the first work that implements this idea for mixing activities.

There is still the need to test the system with a wider audience and more options of mixing systems. One aspect which needs to be checked is the different gain input levels on different analog systems. This requires an automatic gain control on the mixing system side, or at least a pre-processing on the input audio data to fit its levels according to the mixing system characteristics. Different equipments could also provide different frequency responses, requiring the musician to have a feedback of the system output, allowing it to change some parameters to avoid an undesired result and even to take full advantage of the high quality equipment.

We propose the usage of template configurations which can be created to compensate the audio response of an specific system. A template would be used to change the audio parameters on the user input data before processing the job on the server side, to make it the most similar as the user original request. Another possibility is to automate some of the mixing system controls (level, panoramic settings and others), allowing the user to set them remotely.

This work should also be assessed with different target populations and mixing equipments, to validate the main concept and check the feedback of the possible real users of this kind of system. We propose the assessment of this architecture with musicians, professionals and amateurs; professional studio audio technicians; amateur music instrument enthusiasts; ubiquitous music composition users; do it yourself (DIY) audio equipment users.

6. Conclusions

This paper described an architecture to use remote systems for music production and mixing activities. In the proposed access model, different mixing systems are available remotely, managed by a server, which receives job requests from users to execute the mixing activity on the remote systems. The system makes it easier to any musician, amateurs and professionals alike, to test their music production on high quality audio equipment from any online connected device, from smartphones to desktop computers.

The implementation of a working prototype brought some issues which still need to be solved. The option for implementing a complete working prototype, even while being simpler than the complete architectured proposed, allowed us to validate the usage of the system with a few users and reflect upon some important aspects of the solution.

The option to implement a VST plug-in to collect the user audio data from inside the DAW turned out not to be the best approach. Although supposed to be a simple approach, this was not easily configured on different DAWs, due to the fact that the plug-in needs to have multiple input channels. A better solution could be the implementation of a device driver to be configured on the DAW and used to collect the audio data.

The volume level of the individual audio channels also needs to be handled. Depending on the mixing system audio interface and the analog equipment used for the final mixing, some parts of the output audio may be clipped, due to a very high resulting level after the sum of all channels based on the standard level chosen by the user during its individual audio track recording. On the same way, the output level could be very low, and so having a low signal-to-noise ratio on the output audio.

The usage of a DAW on the mixing equipment side, running a script to automate the mixing process is also something that needs to be changed to allow a more generic approach and the usage of several mixing systems, with different equipment models. On the other hand, it was possible to verify the different results when mixing the same input music inside a DAW and on different mixing equipments.

We believe that the proposed architecture contributes by enabling musicians and producers, amateurs and professionals, to use and have access to different equipment available for audio production. Besides, it enables learning experiences and composition activities to be done from any mobile device, being able to request a new music and listen to the results. This avoids the need to be always on the studio or near a desktop computer to interact with this kind of hi-fi equipment.

References

Costa, C.A., Yamin, A.C., Geyer, C.F.R.: **Toward a general software infrastructure for ubiquitous computing**. IEEE Pervasive Comput. 7(1), 64–73 (2008)

Izhaki, R. **Mixing Audio: Concepts, Practices and Tools**, 3rd Edition, Focal Press, 2008

Johann, M. O. **Modifying the ADA8000**. [S.l.:s.n], 2010. Disponível em: http://www.inf.ufrgs.br/~johann/ada8000/

Johann, M. O.; Yefinczuk, A.; Pirotti, R.; Chiaramonte, M. **Análise e Reprojeto de Mesa de Alta Qualidade para Gravação Acústica**. 13º Congresso the Engenharia de Áudio, 19ª Convenção Nacional da AES Brasil. São Paulo, SP, Brasil. 2015.

Kapur, A.,Wang, G., Davidson, P., Cook, P.R.: **Interactive network performance: a dream worth dreaming?** Organised Sound 10(03), 209 (2005). doi:10.1017/S1355771805000956

Keller, D.; Lazzarini, V.; Pimenta, M. S. **Prologue - Ubiquitous Music: A Manifesto**. Ubiquitous Music, pages xi-xxiii, Springer, 2014.

Orto, C., Karapetkov, S.:**Music Performance and Instruction over High-Speed Networks**. Tech. Rep. June, polycom (2011). http://www.polycom.com/global/documents/whitepapers/music_performance_and_instruction_over_highspeed_networks.pdf

Pimenta, M. S.; Keller, D.; Flores, L. V.; Lima, M. H.; Lazzarini., V. **Methods in Creativity-Centred Design for Ubiquitous Musical Activities.** Ubiquitous Music, Chapter 2, pages 25-48, Springer, 2014.

Valladares, T.A.: **Independent Producers: A Guide to 21st Century Independent Music Promotion and Distribution.** Master's Degree Thesis. University of Oregon (2011)

Zawacki, L. F.; Johann, M. O. **Analog Audio Recording Using Remote Servers**. Ubiquitous Music. Pages 83-107. Springer, 2014.