Pulsemeeter - A virtual mixer for linux

Gabriel Rodrigues Chaves Carneiro[1]*, **Flávio Luiz Schiavoni**[1]

[1] ALICE - Arts Lab in Interfaces, Computers, and Everything Else
Departamento de Ciência da Computação
Universidade Federal de São João Del Rei
gabriel.chaves.carneiro@gmail.com, fls@ufsj.edu.br

Abstract. *Pulsemeeter is an application designed to simplify audio management in computer systems. Much like certain mixers group audio sources, the computer's audio system comprises categories such as games, music players, system sounds, and video players, which can be grouped and have their volumes regulated to control the system's sound output. Given that users might engage with multiple applications within these categories simultaneously, this work proposes an abstraction layer that groups similar applications. It aims to offer a more generalized manipulation of system sound sources without requiring constant micromanagement when adjustments are needed. This application aims to streamline the process while retaining the flexibility for specific adjustments if required. Notably, it seeks to provide a viable alternative to virtual mixers like Voicemeeter, specifically tailored for Linux systems. The application's focus lies in simplifying audio control without compromising on versatility, targeting improved user experience within the Linux environment.*

Keywords — mixer; linux; audio;

1. Introduction

In the realm of audio production, the evolution of technology has introduced virtual mixers, providing software-based solutions that emulate the functionalities of physical mixers. These virtual mixers, simulate the traditional mixer's controls and capabilities within a computer interface. They offer a wide array of tools for manipulating audio, from basic volume adjustments and equalization to more advanced effects and routing options.

In this paper, we present Pulsemeeter[1], a Free and Open-Source Software (FOSS) dedicated to managing system audio in Linux environments. Resembling traditional mixers with standard controls like volume sliders, panning knobs, and equalization, Pulsemeeter's defining feature revolves around the concept of 'groups.' These groups redefine the conventional approach to audio manipulation within mixers and act as a higher-level abstraction, allowing users to bundle multiple audio sources together, thereby treating them as a unified entity for control purposes.

The concept of groups within a mixer isn't novel in the realm of Linux audio manipulation. Other tools offer similar functionalities, enabling users to manage multiple audio sources collectively. What distinguishes Pulsemeeter from others is its capacity to abstract and simplify the process. It streamlines the creation and management of these groups, offering a more user-friendly interface and easing the complexities associated with handling diverse audio streams.

This paper seeks to delve into Pulsemeeter's functionality, focusing on its use of group abstraction in simplifying audio control within the Linux ecosystem. By elucidating the core concept of groups and exploring how Pulsemeeter abstracts this functionality, this study aims to highlight the software's contribution in providing a more accessible and user-friendly approach to managing system audio in Linux, despite the availability of similar functionalities in other tools.

*Supported by CAPES.

2. Related Work

The concept of virtual mixers isn't new; they've been around for a while, offering diverse implementations suited to different needs. These digital tools, integral for managing and shaping audio signals, have seen significant evolution. They range from basic mixing consoles to sophisticated software solutions, each tailored for specific applications. Some excel in live sound engineering, enabling real-time adjustments for concerts and events, while others cater to studio production, providing intricate controls for precise audio refinement. Each variation of these virtual mixers brings unique functionalities, serving distinct scenarios and user preferences. Whether you're a live sound technician or a studio producer, there's a perfect match available to streamline your work and ensure top-notch sound quality.

In this session, we'll be diving into the diverse implementations of virtual mixers, and the concepts involved. Our main focus will be exploring these different models, looking at their features, functionalities, and how well they suit specific uses. The aim here is to break down these various versions of virtual mixers, highlighting what they do best, their limitations, and the scenarios where they truly excel. By exploring the concepts involved and different virtual mixers, we can gain a better understanding of Pulsemeeter's specific use case and its relevance within this domain. Understanding how various virtual mixers are utilized helps illuminate how Pulsemeeter fits into this landscape, highlighting its distinctive functionalities and its role among these tools.

2.1. Mixers

In traditional mixers, audio control primarily revolves around managing individual audio channels. These channels correspond to different audio sources, each possessing its own set of controls, typically including volume faders, panning knobs, and equalization parameters. Users can manipulate these controls independently, tailoring the characteristics of each audio channel to suit their preferences.

However, the concept of groups in mixers represents a departure from this individual channel-centric approach. Groups act as a higher-level organization tool, allowing users to combine multiple individual channels into a collective unit. Once combined, these channels function as a unified entity within the mixer.

The creation of groups facilitates simultaneous control over multiple related channels. Users can adjust parameters like volume levels, equalization settings, and routing configurations for the entire group as a whole. This collective control simplifies the management of multiple audio sources that are logically linked or intended to function together.

Figure 1: An audio mixer with four groups configuration

Figure 2: VoiceMeeter Banana

For example, in a live audio setup, several microphones might be grouped together to collectively adjust their overall volume levels while maintaining their relative balances. In a recording studio scenario, instruments belonging to the same section, such as drums or vocals, could be grouped to apply consistent equalization settings across the entire section.

Groups in mixers streamline the workflow by offering a more efficient way to handle complex audio configurations. They reduce the need for repetitive adjustments across individual channels and enhance the ability to manage related audio sources collectively, contributing to a more coherent and synchronized audio output.

2.2. Voicemeeter

Voicemeeter[2] is a proprietary donationware software, exclusively available for the Microsoft Windows operating system, offered in three versions: Voicemeeter, Voicemeeter Banana, and Voicemeeter Potato. The only difference among them is the supported number of inputs and outputs. The Potato version requires activation after a trial period, while the other two allow free usage.

All three versions aim to provide a virtual audio mixer capable of individually mixing each device, selecting multiple audio outputs for inputs, whether they are physical inputs like a microphone or virtual inputs like music. This capability enables scenarios like transmitting microphone audio to headphones while playing music through both headphones and speakers, with individual volume control for each.

Voicemeeter also supports the use of MIDI devices to execute commands. Users can map buttons to perform necessary functions, including control commands such as volume adjustments.

2.3. Pulseaudio

PulseAudio[3], a commonly used sound server for Unix-like operating systems, including various Linux distributions, plays a crucial role in managing audio on these platforms. Serving as a middle layer between applications and the audio hardware, PulseAudio enables advanced audio functionalities, allowing multiple applications to share and control audio output concurrently.

One of PulseAudio's primary strengths lies in its capacity to handle complex audio routing. It offers a flexible and configurable system that lets users direct audio streams to specific hardware devices or software sinks, making it easy to switch between different input and output sources. This capability enables users to, for instance, route audio output to external speakers, headphones, or virtual audio devices seamlessly.

Furthermore, PulseAudio incorporates features for volume control[4], enabling users to adjust volume levels for individual applications or audio streams independently. This precise control over audio levels ensures a more customized and balanced audio experience across various concurrently running applications.

Moreover, PulseAudio provides tools for dynamically managing audio streams. It supports features like automatic routing based on device availability, ensuring smooth transitions when connecting or disconnecting hardware devices. This adaptability enhances user experience by ensuring uninterrupted audio output while switching between different audio devices or configurations. Overall, PulseAudio's comprehensive set of features and its adaptability make it a fundamental component of audio management on Linux-based systems, providing a versatile framework for handling audio tasks.

2.4. Jack Audio Connection Kit

The Jack[5] Audio Connection Kit (Jack) stands as a pivotal tool within the realm of Linux-based audio management. Developed with a focus on low-latency audio connections, Jack serves as a sophisticated audio infrastructure facilitating the seamless routing of audio signals between applications and audio devices in real-time.

At its core, Jack operates as a low-latency audio server, offering a robust framework for handling audio connections with unparalleled precision. Its design revolves around a client-server architecture, where audio applications function as clients, utilizing Jack as the intermediary for audio signal transmission. This architecture ensures precise synchronization and minimal latency, making Jack particularly suited for professional audio production environments, live performances, and intricate sound manipulation tasks.

One of Jack's distinguishing features lies in its ability to support complex audio configurations. It allows users to create intricate audio routing setups, connecting multiple applications, virtual instruments, and hardware devices in a flexible and customizable manner. This level of versatility enables musicians,

audio engineers, and enthusiasts to design intricate audio workflows tailored to their specific needs, whether it involves complex studio setups or live performance environments.

2.5. Pipewire

PipeWire[6], an advanced multimedia framework developed for Linux-based systems, operates as a multimedia processing server, offering an extensive range of functionalities. Positioned as a successor to both PulseAudio and JACK[7], PipeWire aims to unify various multimedia needs by providing a singular framework capable of handling diverse audio and video use cases[8].

At its core, PipeWire facilitates seamless multimedia integration and management across applications and devices. It works as a central hub, allowing applications to share audio, video, and other multimedia streams efficiently. This framework is designed with a modular architecture, enabling it to adapt to a variety of multimedia requirements while ensuring compatibility with existing systems and software.

One of PipeWire's significant advantages lies in its ability to handle diverse audio and video workflows simultaneously. It streamlines the management of multimedia tasks, providing low-latency processing suitable for both consumer and professional-grade applications. Moreover, PipeWire's versatility extends to supporting advanced features such as screen capturing, screen sharing, and device management, offering a comprehensive solution for multimedia handling within Linux environments.

3. Developing Pulsemeeter

The inception of Pulsemeeter's development began with a comprehensive exploration of user challenges in managing diverse audio sources across multiple applications within the Linux environment. This preliminary phase involved an extensive analysis aimed at identifying an approach that could simplify audio control without compromising the necessary flexibility for specific adjustments.

At the core of Pulsemeeter's development was the implementation of an abstraction layer—an essential addition facilitating the grouping of audio sources from similar applications, using only native linux APIs[9]. This strategic layer aimed to introduce an intuitive interface for regulating system sound without requiring continuous micromanagement. Operating as a categorized and regulated framework, this layer allowed for the bundling of audio sources within defined groups while maintaining the capability for nuanced modifications when needed.

Guiding Pulsemeeter's evolution was an iterative development approach underscored by active community engagement and continuous feedback integration. Through numerous iterative cycles, the application underwent rigorous testing and refinement, harnessing diverse inputs from the community to significantly shape and enhance the application's features and usability within the Linux environment.

An integral phase encompassed a comprehensive comparative analysis, primarily focusing on Pulsemeeter's functionalities concerning existing virtual mixers, particularly Voicemeeter. This analysis aimed to ascertain and underscore Pulsemeeter's distinctive capabilities within the Linux ecosystem, emphasizing its pragmatic advantages over existing solutions.

To substantiate Pulsemeeter's efficacy, a series of meticulously devised use case scenarios were crafted to simulate real-world usage scenarios. These scenarios formed the foundation for extensive usability testing, enabling users to interact with the application, provide critical feedback, and assess its practicality in efficiently managing system sound sources.

4. Results

In the hardware realm, Pulsemeeter deals with traditional audio devices—microphones, speakers, headphones, and other peripherals. It introduces the concept of virtual devices, offering users the capability to create custom groups that simulate these physical components. These virtual devices aren't limited by hardware constraints; they're flexible and configurable entities, as seen in image 3. For example, users can amalgamate multiple microphones into a single virtual device for a unified input source or aggregate speakers into a virtual grouping for synchronized audio output.

However, Pulsemeeter goes beyond merely managing hardware. It recognizes and interacts with software applications as integral components contributing to the audio landscape. Each application generates its own audio stream, allowing Pulsemeeter to control their input and output destinations. Pulsemeeter's strength lies in its capability to harmonize diverse components—hardware inputs, hardware outputs, and software applications—into unified virtual configurations. These composite virtual devices embrace a wide range of audio sources, enabling seamless interactions and control. For instance, users can forge a virtual device that mixes inputs from multiple microphones, audio output from a music player app, and the audio stream from a video conferencing tool. These combined audio streams can be directed not only to designated headphones or speakers but also channeled into other applications. This adaptability offers an extensive array of output possibilities and configurations, empowering users to craft personalized audio environments with remarkable versatility.

This inherent flexibility allows users to construct complex audio pathways. It enables routing audio streams from various sources—be it hardware inputs, software applications, or a hybrid combination—through customized configurations of virtual devices. Users can tweak settings, apply effects, and define unique routing paths for each virtual device, tailoring their audio environment to suit specific needs.

In summary, Pulsemeeter operates as an integrative hub orchestrating a dynamic interplay between hardware inputs, hardware outputs, and software applications' audio streams in the Linux ecosystem. Its capacity to merge and manage these diverse elements into adaptable virtual constructs fosters a malleable and personalized audio environment catering to a wide spectrum of user demands and preferences.

The development journey of Pulsemeeter exemplifies the power of community collaboration in shaping open-source projects. What began as an experimental concept saw significant alterations thanks to community involvement. Multiple revisions led to a departure from its initial purpose, resulting in a more adaptable audio control system.

Pulsemeeter's current version presents notable improvements. Users now have the freedom to create and modify virtual audio devices with custom names, enabling a higher degree of personalization. Its capability to fine-tune device assignments for specific applications grants users precise control over their audio preferences.

The project's evolution highlights the impact of collab-

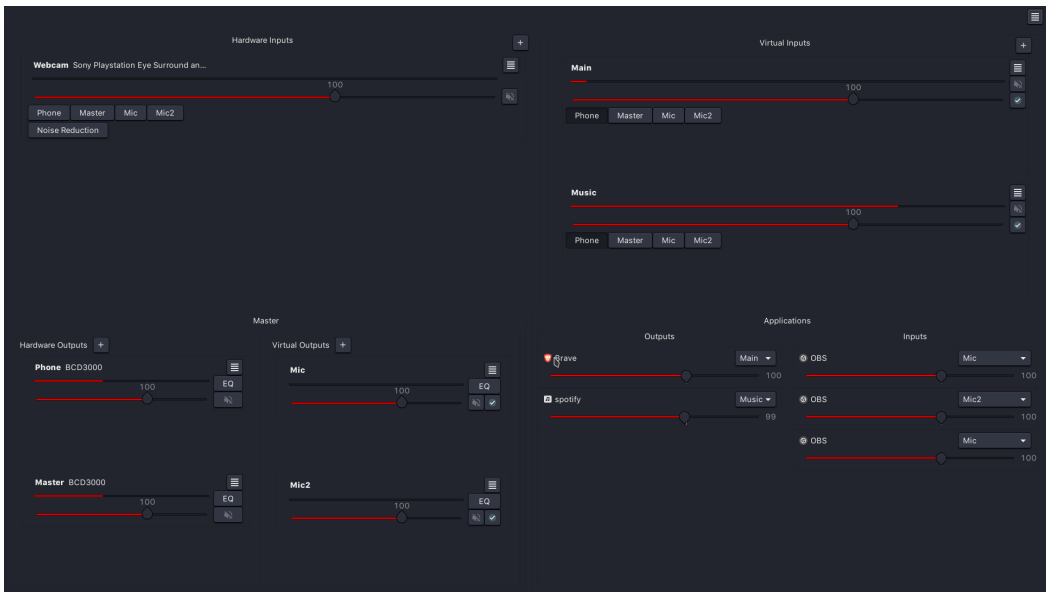

Figure 3: A typical figure

orative engagement and iterative development. The diverse community inputs fueled innovation, ultimately producing a product that exceeds its original goals. Pulsemeeter serves as a testament to the growth of open-source software, emphasizing the significance of community participation and adaptability in its evolution.

In summary, Pulsemeeter's development illustrates the journey of an open-source project, transitioning from an experimental idea to a more adaptable and user-centered solution for managing audio in Linux. Its story underscores the vital role played by community involvement and continuous improvement in shaping open-source software.

5. Conclusion

Pulsemeeter's prowess stems from its foundation as an open-source endeavor, nurtured by the active engagement of a diverse and vibrant community within the Linux environment. This collaborative ethos has been pivotal in shaping the evolution of Pulsemeeter, transforming it into a feature-rich solution through the concerted contributions of developers, enthusiasts, and users alike.

Originally meant to mirror Voicemeeter, Pulsemeeter eventually took a different path, providing users with more freedom and flexibility. Departing from its initial clone concept, Pulsemeeter evolved uniquely, introducing features that allowed greater customization and control, empowering users beyond its original scope.

Despite boasting a rich array of features, Pulsemeeter is far from a finished product. Its continuous evolution is fueled by the ongoing collaborative efforts of the community. The commitment remains steadfast in addressing user needs, fine-tuning existing functionalities, and charting new territories in Linux-based audio control.

Looking ahead, the community's engagement stands as a cornerstone for Pulsemeeter's trajectory. The collective endeavor is laser-focused on fortifying its capabilities, elevating the user experience, and accommodating an even broader spectrum of audio control requisites. Pulsemeeter stands as an embodiment of community-driven progress, exemplifying how collaboration is not just advantageous but pivotal in advancing Linux-based audio management.

In essence, Pulsemeeter's journey exemplifies the strength derived from collaborative development, emphasizing the significance of ongoing engagement and shared contributions within the Linux ecosystem. Its future strides are poised to echo the collective aspirations and evolving needs of its diverse user base, further cementing its position as a community-driven solution for refined Linux-based audio control.

6. Acknowledgment

Authors would like to thanks to all ALICE members that made this research and development possible. The authors would like also to thank the support of the funding agencies CNPq, CAPES and FAPEMIG.

References

[1] Carneiro, Gabriel Rodrigues Chaves. *Pulsemeeter: um mixer virtual*. Monografia (bacharelado em ciência da computação), Universidade Federal de São João del-Rei, 2022.

[2] VB-Audio. Voicemeeter.

[3] FREEDESKTOP. Pulseaudio.

[4] Victor Gaydov. Pulseaudio under the hood. *Victor Gaydov*, 2017.

[5] Paul Davis. Jack audio connection kit.

[6] FREEDESKTOP. Pipewire.

[7] Wim Taymans. Pipewire-jack. `https://gitlab.freedesktop.org/pipewire/pipewire/-/wikis/JACK`, 2021. Acessado: 2022-12-01.

[8] Patrick Louis. Making sense of the audio stack on unix. 2021.

[9] Flávio Luiz Schiavoni, Antonio José Homsi Goulart, and Marcelo Queiroz. Apis para o desenvolvimento de aplicações de áudio. *Seminário Música Ciência Tecnologia*, 1(4), 2012.