Audio Engineering Society

Convention Paper 8081

Presented at the 128th Convention
2010 May 22–25 London, UK

Audio latency measurement for desktop operating systems with onboard soundcards

Yonghao Wang[1], Ryan Stables[2], and Joshua Reiss[3]

[1] Centre for Digital Music, Queen Mary, University of London, London, E1 4NS, UK
yonghao.wang@elec.qmul.ac.uk
and Birmingham City University, Birmingham, B4 7XG, UK
yonghao.wang@bcu.ac.uk

[2] Birmingham City University, Birmingham, B4 7XG, UK
ryan.stables@bcu.ac.uk

[3] Centre for Digital Music, Queen Mary, University of London, London, E1 4NS, UK
josh.reiss@elec.qmul.ac.uk

ABSTRACT

Using commodity computers in conjunction with live music digital audio workstations (DAW) has become increasingly more popular in recent years. The latency of these DAW audio processing chains for some application such as live audio monitoring has always been perceived as a problem when DSP audio effects are needed. With "High Definition Audio" being standardised as the onboard soundcard's hardware architecture for personal computers, and with advances in audio APIs, the low latency and multi-channel capability has made its way into home studios. This paper will discuss the results of latency measurements of current popular operating systems and hosts applications with different audio APIs and audio processing loads.

1. BACKGROUND

Over the past 20 years, computer music has shifted from the mainframe to DIY culture with miniaturised and democratised live applications [1]. The increasing power of the personal computer drives the recent tendency towards using commodity computer based digital audio workstations (DAW) in live performance or recording environments. There are obvious advantages of using computers, which can be flexibly configurable with abundant software packages and plug-ins to replace or emulate some cumbersome hardware devices.

However the latency of the DAWs has always been perceived as a problem for some real-time audio applications. The constraint of maximum allowed latency in audio processing varies between different applications. In audio streaming over a packet switched network, the one-way delay can be at the magnitude of seconds, and still be regarded as real-time [1]. In live performance and record monitoring environments, the maximum tolerable delay is around 10ms to 30ms depending on the different environments of performers and instruments [3][4]. For some performers, such as saxophone players, the threshold is even lower. Recent comprehensive testing results can be found at [5]. In the digital audio chain for live music, the DSP and software monitoring platform seems to be the main cause of latency [6].

Professional digital consoles normally have overall system latency no more than 2 ms. There are concerns that surround the use of computer based DAWs for low latency work (less than 10ms) due to unexpected jitters in sound when the CPU is heavy loaded [7]. Therefore, professional audio interface cards provide hardware based monitor sub-mixing or bypass routing for the purpose of offloading the CPU.

In 1998, researchers presented the results and discussed the causes of audio process latency of common operating systems [8]. It was suggested that the ideal latency time could be 3ms, and revealed the difficulties involved in achieving this. In 2001, research [9] indicated that the proper architecture of audio API stacks should keep the latency in the audio processing path constant without being affected by heavy CPU load tasks. The most promising low latency audio layers at that time were Linux ALSA (Advanced Linux Sound Architecture) and Mac OS X CoreAudio. At that time the tasks used in order to cause CPU load were not from audio dependent applications.

Audio driver architecture has evolved over the years, along with live audio applications and hardware platforms. The adaptive audio effects [10] which use feature extraction to create control signals for the processing of sound have often been proven to have high computational cost, leading to heavy CPU loads. However with the appropriate side-chain design, multi-threading support from audio host platform and the concurrency of the software architecture, the hypothesis can be made that the intelligent subsystem and multiple audio processing paths should not affect the real time audio processing path even when the CPU load is coming from the audio application itself.

2. TESTING METHOD

The sound source can be constructed mathematically in the form of either a single pulse or pulse train. When playing back the sound source, it is split into two channels. One channel is sent directly to recording devices, bypassing the operating system and the second is routed through the test system and recorded as a second channel using the same recording device. The recording device can be a digital recorder or a computer with professional sound interface. By analysing the final recording, the latency of audio processing path can be determined as shown in Figure 1.

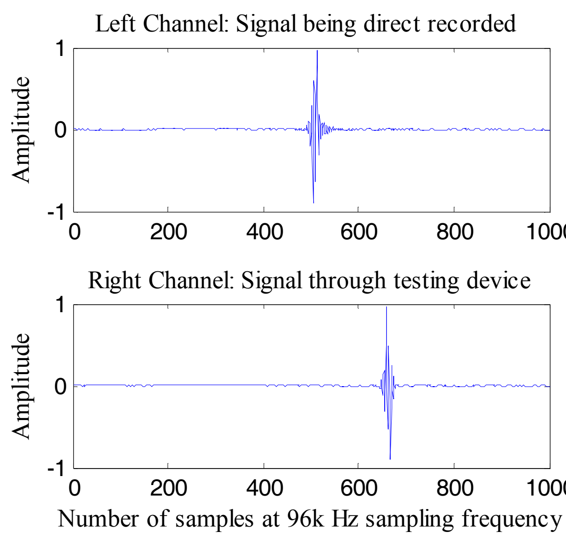

Figure 1 Latency measurement from recorded signals

The single pulse was used to access the minimum latency we could possibly achieve, whereas the pulse train was used for testing the glitches, variable latency and loss of information.

The capability of adjusting software buffers needs to be considered in order to make them comparable for different test cases.

2.1. Test plan

Overall, there are many combinations of host DAWs, operating systems, driver APIs, soundcards and hardware. Therefore a carefully designed test plan is needed that contains a set of test cases in order to verify specific aspects of system latency by fixing and altering variables in the test domain.

In addition to this, the work presented in the study contains some special conditions for the purpose of cross-reference. These included latencies of commonly used professional digital consoles and DSP development hardware.

The focus of our testing is the commodity personal computer. Therefore the results are taken mainly from the machine’s onboard soundcard. For the purpose of comparison however, external soundcards were also included in our testing procedures.

In general, four different testing groups were set:

- Test case 1 – Vanilla test, the purpose of this test is to obtain general latency results for different operating system and host combinations with exhaustive available software hosts. We tested common software such as Audacity, Logic Pro, Ableton Live, and Ardour.

- Test case 2 – Stress test, based on the results from the “Vanilla test”, we selectively tested the latency of our chosen hosts with a heavily loaded CPU.

- Test case 3 – Adaptive effect test, to test if the audio processing latency is affected when the CPU load comes from the audio application itself, especially when the host handles multichannel audio and the adaptive audio effects are actively being used.

- Test case 4 - Cross-reference test. The purpose of this test is to get latency measurements from various systems other than common operating systems with onboard soundcards in order to avoid bias when evaluating the results of the above three tests. The tests include digital consoles, external soundcard, and the audio development hardware.

2.2. Variables

The variables of all the test cases were comprised with hardware, operating systems, and host applications. Ideally, the same hardware platform installed with multiple operating systems was used wherever possible. Different hardware platforms were tested as cross-reference for separating the hardware performance influences from that of the operating system.

2.2.1. Hardware platforms

The Intel based Apple computers are used as the main test platforms as they are able to support all three popular operating systems with additional configurations. A common PC laptop with similar hardware specification was also tested in order to verify the validity of test results taken from the Apple computers when the operating systems other than Mac OS X installed.

The main component of on-board sound system is the hardware audio codec. Both ALC885 and CS4206A codec chips are in compliance with Intel HD audio, which support multiple inputs and outputs channels with sampling frequency up to 192k Hz [11][12]. Table 1 lists the details of computer platforms, in which the CPUs are Intel Core 2 Duo with different CPU clock speed.

Made	CPU Speed (GHz)	Memory (GB)	Sound card Codec
iMac	2.66	2	ALC885
Mac Book Pro	2.4	2	ALC885
Mac Book Pro	2.8	4	CS4206A

Table 1 Hardware platform of the test systems

For cross-reference testing, the tested devices and hardware listed in the Table 2.

Type	Made
Digital Console	Yamaha 01v
Digital Console	Yamaha O2R 96
Digital Console	Yamaha DM2000
SHARC Board	ADSP-21161N EZ-KIT
USB soundcard	M-Box 2 Mini

Table 2 Cross-reference testing devices

2.2.2. Operating Systems

Table 3 lists the operating systems tested. The Windows and Linux operating systems tested are all 32-bit versions. All operating systems updated with latest patches.

Operation System	Short Name
Apple Mac OS X 10.5.8	OSX (Leopard)
Apple Mac OS X 10.6.2	OSX (Snow Leopard)
Microsoft Windows XP	WinXP
Microsoft Windows 7	Win7
Ubuntu Linux 9.10	Linux

Table 3 List of test operating systems

The Linux Operating system Ubuntu 9.10 has "Ubuntu Studio Audio Package" installed which contains the real-time preemption kernel patch for the 2.6.31 kernel.

One of the most important components within operating systems is the audio Application Programming Interfaces (APIs). They play the important roles in relation to audio processing latency to provide the middle layers between the low level sound system and the high level software applications. The default APIs of our tested operating systems are listed in the Table 4.

API	Platform	Short name
Microsoft DirectSound & DirectSound Capture	Windows XP, Windows 7	DirectSound
Microsoft Multimedia Extensions	Windows XP, Windows 7	MME
Apple CoreAudio	Mac OS X	CoreAudio
Advanced Linux Sound Architecture	Linux	ALSA

Table 4 list of audio APIs

There are additional APIs which are used by some audio applications but not included by defaults by operating system, such as Steinberg Audio Stream Input Output (ASIO), PortAudio [13], and JACK API [14]. Each of them serves different application purpose.

2.2.3. Audio application hosts

In order to test the audio processing latency of operating systems, the software is needed to capture the audio signal and playback it.

Rather than using a simple "play through" code, in most test cases, the completed DAW hosts were tested, because the goal of test case 3 is to test whether audio latency is affected by multichannel audio processing and intelligent audio effects. The host software provides the facilities to be able to carry out this test. Table 5 lists the hosts we used in the testing.

Hosts code	Host name	Notes
1	Apple Logic Pro 8.0	
2.a	Ableton Live 8.1.1	with Max/Msp
2.b	Ableton Live 8.0.1	
3.a	Audacity 1.2.5	Stable version
3.b	Audacity 1.3.11	Beta version
4	Ardour 2.8.7	Version 2.8.2
5	CAPlayThrough	Play through code

Table 5 List of test Audio Hosts

2.2.4. The limitations of test plan

The latency measurements were tested based on the popular operating systems installed in common Apple computers in combination with onboard soundcards. The range of different computer hardware is limited, however, given that the commodity computer architecture is fairly standard, the computers we tested are common platform for DAWs. The results should be interesting in some aspects.

The second limitation is the limited number of external soundcards we tested. The results however should still be valid for showing the performance of operating systems performs with onboard soundcards and default APIs.

The third limitation is that there are very few audio application which supports all different operating systems. Perhaps the portability of audio applications and the optimisation of using native API and operating system features are the two conflicting efforts for software development. Therefore the cross-platform application such as Audacity uses the middle layer API "portaudio" to unify the audio programming interfaces for different operating system platforms.

The matrix of host applications and supported operating systems are listed in the Table 6.

Host	Windows	Linux	Mac OS X
1	No	No	Yes
2.a	Yes	No	Yes
2.b	Yes	No	Yes
3.a	Yes	Yes	Yes
3.b	Yes	Yes	Yes
4	No	Yes	Yes

Table 6 Matrix of Hosts and Operating systems

3. TEST RESULTS

3.1. Vanilla Test

In this test, we try to obtain the general picture of latency over our various audio hosts and operating systems with the built-in onboard sound systems and default settings. Table 7 shows the latencies measured using Audacity in different platform:

Host	OS	APIs	Latency (ms)
3.a	MacOSX	CoreAudio	19
3.a	WindowsXP	MME	257
3.a	Windows7	MME	244
3.b	MacOSX	CoreAudio	30
3.b	WindowsXP	MME	398
		DirectSound	152
3.b	Windows7	MME	399
		DirectSound	201

Table 7 Latency of Audacity host with sampling frequency 44100 Hz

We tested two versions of Audacity, the stable version 1.2.5, which uses "portaudio v18" and the beta version 1.3.11, which uses "portaudio v19". The "portaudio" library provides cross-platform audio API interfaces with encapsulation of platform dependent APIs such as CoreAudio, ALSA, DirectSound, or ASIO.

In Audacity 1.2.5, no buffer settings are available for end user, whereas in Audacity 1.3.11, the recording audio buffer is set to zero. No special drivers were installed for Windows platforms.

In Ubuntu Linux 9.10, the current software playback function of Audacity has some problems with newly adopted PulseAudio sound server system. It is considered that further testing using other Linux distribution is needed.

Vanilla test identified some low latency hosts for further test groups. The Table 8 shows the latency measurements of these hosts. The buffer settings are either the lowest that hosts applications support or the lowest at which monitored incoming sound can be recorded.

Host	OS	API	Buffer[1]	Latency (ms)
1	OSX	CoreAudio	32*2	5
2.a	OSX	CoreAudio	14*2	4.2
2.b	WinXP	DirectX	512	73
2.b	Win7	DirectX	512	81
4	Linux	ALSA	64*2	3.3
4	OSX	CoreAudio	32*2	6.2

Table 8 low latency hosts with sampling frequency 44100 Hz

Table 9 shows the lowest possible latency in different operating system we can possibly get by using highest sampling frequency at 96k Hz supported by onboard soundcards.

Host	OS	API	Buffer	Latency (ms)
2.b	WinXP	DirectX	512	73
4	Linux	ALSA	64*2	1.68
4	OSX	CoreAudio	32*2	3.54

Table 9 lowest latencies from the Villain test

In addition, the Vanilla Test found the latency measurements taken from Mac OS X 10.5.8 Leopard are almost identical to Mac OS X 10.6.2 Snow Leopard. And there are similar results for Windows system whether installed on an Apple computer or on a PC laptop with similar hardware specification.

3.2. Stress test

The audio processing latency caused by CPU stress is tested rather than by the I/O stress. With advanced DSP techniques being widely used in real-time audio

[1] Displayed buffer is in number of samples, where multiplication of 2 indicates two-way buffers.

processing, the computational cost is more likely to be CPU stressed tasks.

The sound source, consisting of a series of pulses at constant intervals is used as test signal. It is observed that even without CPU stress, when the latency is less than 5 ms, the audio signal suffers from distortion and glitches and loss of infomration.

The following table shows the latency with and without CPU load. System monitoring software is used to ensure the CPU load outside audio application is 100%. In addition, the host software normally has built-in CPU meter to indicate the CPU load of audio processing only [15]. This means that even when the system monitor indicates 100% CPU load, the audio application CPU load may still be very low. It is observed, however, that if the audio processing CPU load increases, it is reflected on the outside system monitor meter.

Host	OS (API)	Without Load	With Outside load	With Audio load
2.b	WinXP (DirectX)	73ms (buffer 512)	81ms (buffer 512)	104ms (buffer 512)
2.a	OS X (CoreAudio)	4ms (buffer 14*2)	4 ms (buffer 14*2)	5.80ms (buffer 14*2)
4	Linux (ALSA)	3.31ms	3.31ms	error
		22ms (buffer 512*2)	22ms (buffer 512*2)	22ms (buffer 512*2)

Table 10 Audio latency with different CPU loads

The tests results indicate that the CPU load generated outside of the audio applications have very little effect on the latency of the audio processing chain. For Mac OS X and Linux systems, this effect cannot be observed, for Windows system, this is very small.

To some extent, when the CPU load comes from inside of the audio application, latency is increased by 1-2ms for Mac OS X.

In Linux system, it causes a system error when Ardour tries to connect to the JACK audio server. With an increased audio buffer setting to 512 samples in the Linux system, the inside CPU load doesn't seems affect the latency.

However, with high audio processing load, test signal being the pulse train, it was observed that the signal suffers distortions or loss of pulses as shown in the Figure 2.

The Figure 2 show in Linux, the signal channel processed by operating system lost some information when the buffer setting and sampling frequency were set in order to obtain very low latency. The similar effects were observed in Windows and Mac OS X systems.

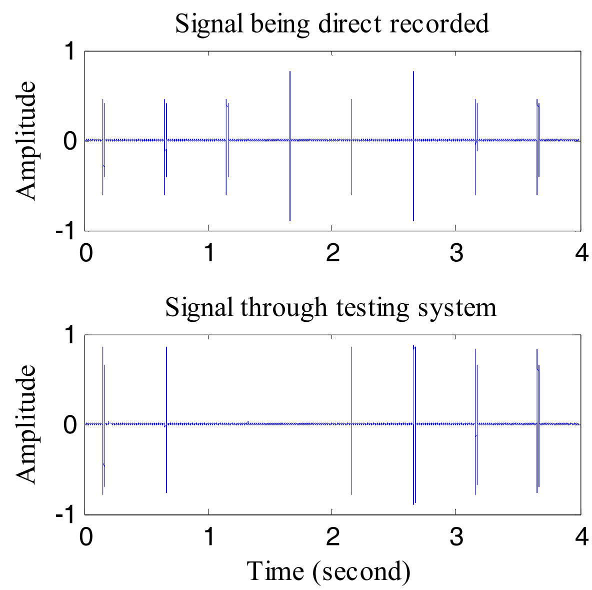

Figure 2 Loss of pulses in audio processing path at low latency setting

3.3. Multichannel latency

The Table 11 shows the latency variation caused by large number of channels e.g. over 50 channels. This is the same effects to that of increased internal CPU load.

Host	OS	Latency Signal Channel	Latency Multi Channel
2.b	WinXP	73 ms	104 ms
2.a	OS X	4 ms	5.80 ms
4	Linux	3.31 ms	error
		22 ms	22 ms

Table 11 Multichannel latency effects

3.4. Adaptive Effect latency

Adaptive audio effects combine audio feature extraction and audio processing in order to give musician and audio engineer anther creative dimension. This helps in generating new musical concepts and contributes to making existing tasks and processes more intelligent. In the real-time multichannel mode, it may require the audio analysis subsystem to synchronise with the audio processing chain in order to make audio effect decision, which can be computational cost if the large number of channels are involved and the required analysis rate is high.

The measurement in Table 11, however, did not include the adaptive audio effects. The current audio application hosts have not widely supported this type of audio effects yet. The "Max for Live" functionality of Ableton combines Max/Msp with Ableton Live's plug-in structure, providing an interesting starting point. Adaptive audio effects are created fairly easily using "Max for Live". It is of interest to test the latency in this configuration.

In order to obtain an undistorted audio signal, the buffer is set to 256 samples. The effect plug-in is based on feature extraction created to obtain "loudness", "brightness", "noisiness", and "onsets" of audio signal. Based on these features, the amplitude of the signal is modulated with some random parameters. This patch is then applied to multiple channels in order to increase the internal CPU load of the host. The test shows the latency performance has a vast difference when Max/Msp edit window is opened and closed.

Host	OS	Max Window	Single channel (ms)	Multi Channel (ms)
2.a	OS X	Opened	97-99	100-103
2.a	OS X	Closed	32 -51	39-67

Table 12 Latency measurement of "Max for Live"

3.5. Cross-reference Test

Table 13 shows the latency measurement of dedicated hardware audio devices.

Type	Latency
Yamaha 01v	2.42 ms
Yamaha O2R 96	2.04 ms
Yamaha DM2000	1.99 ms
ADSP-21161N EZ-KIT (SHARC)	1.60 ms

Table 13 Latency of dedicated digital audio hardware

Table 14 shows the latency measurement using an external soundcard and dedicated ASIO soundcard driver for Windows. Under this circumstance, the Windows platform performs at a comparable level to Mac OS X with same buffer setting. Though, the Mac OS X supports lower buffer settings up to 6.8ms.

Host	Platform	API	Buffer setting	Latency (ms)
2.a	MacOSX	CoreAudio	128*2	11.9
2.b	WinXP	ASIO	128*2	12

Table 14 Latency measurement of external soundcard M-box 2 mini

The Mac OS X CoreAudio driver has also been patched by manufacturer to support this particular soundcard.

4. DISCUSSION

4.1. Overall latency pictures

Beginning with the cross-platform host Audacity, the Vanilla Test obtained the general latency picture of operating systems with onboard soundcards.

It shows that the latency of record enabled monitoring of beta version of Audacity is actually worse than the older stable version. This might link with the regression report of using newer "portaudio v19" library (according to Audacity development website). In addition, the Audacity software used in this testing are pre-built binaries. Giving its open source nature, to test again with compiling "portaudio" and Audacity from source code to take advantage of native audio API could be further investigated

With the onboard Intel HD audio sound system, the Linux and Mac OS X operating system have low latency performance, and windows DirectSound API performs better than its legacy MME sound API.

With native supported sound driver APIs, the audio hosts dedicated for live application could have low latency within 8-10 ms monitoring requirements.

In 2001, [9], the lowest measured latency was 2.72 ms. It was measured from Linux system with ALSA audio API that replaced default OSS at that time.

ALSA has already become the default Linux audio driver. Our test results show that the lowest latency is provided by open source DAW project Ardour in Linux with ALSA sound driver and JACK audio connection. When using sampling frequency at 96k Hz, the measurable latency can be as low as 1.68 ms (see Table 9). This is comparable with the Yamaha digital consoles tested in section 3.4 (see Table 13).

Another observation is that the reported latencies of most software hosts do not match the measured values. The only exception is Ardour with Linux systems.

Figure 3 shows the plotting of latency measurements and the latency reported by hosts according to buffer settings. The measured value and software reported values are consistent for Ardour in Linux. It needs further research to confirm if the real time Linux kernel helped with audio host maintains accurate timing information and scheduling.

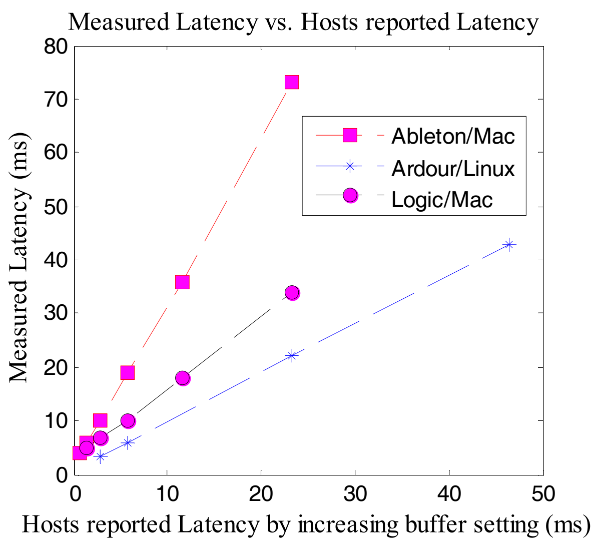

Figure 3 Measured Latency vs. Hosts reported Latency in millisecond with different buffer setting

It is note that according to the results of cross-reference test, the measured latencies did match the reported latencies for Mac OS X system if an external soundcard is used.

With the external soundcard and driver being used, the Windows system could have comparable low latency as Mac OS X when using the same buffer setting.

4.2. Latency under load

In [9], it was shown that the CPU load outside the audio application had little effect on the latency of audio processing.

In our research, the effects on audio processing latency by CPU load caused by audio application itself are evaluated. Consistent with [9], the CPU load outside audio application has unnoticeable effect on the latency for Mac OS X and Linux System, whereas the CPU load inside audio application has caused some small increases of latency generally for all operating systems.

However, the research in [9] did not mention the quality of audio signal when the low latency is required the CPU is stressed. Our research indicates that in low latency mode, especially with CPU is stressed by internal audio processing load, the signal suffers losses and distortion.

It is worth noting that the hardware architecture of the SHARC board is fairly similar to that Intel HD-Audio architecture [16]. However it has very low latency (see Table 13) with good signal quality. The measurement is taken by running an embedded “talk through” example code. This embedded software is driven by hardware level interrupt with enabled DMA features. The software architecture of this is quite different with computer based sound system.

4.3. Multichannel and Adaptive audio effects

The increasing number of channels alone seems do not cause the increasing of audio processing latency. Only when the channel number is increased considerably to around 50 audio channels, it does affect the latency in the same way as increasing the internal CPU load from the audio application host (see Table 11).

The adaptive audio effects provide new creative dimension and intelligent workflow. Use advanced feature extraction based audio processing in real-time is proven interesting and challenging. However the current audio application hosts have not been able to support it widely and flexibly, with the exception of side chain based plug-ins etc. Therefore the test is limited by the available host and the way the host operates.

The “Max for Live” product supports this flexibility by incorporating a Max patch as plug-in. However the results show that the variations of latency do not strongly correlate with audio processing load. The variations of latency might be caused by the configuration and software structure themselves.

5. CONCLUSION

5.1. Summary

This research demonstrated and discussed the test results of the real-time audio processing latency of current popular operating systems with onboard soundcards. To the best knowledge of the authors, the most recent research on this subject was in 2001 [9].

In addition to updating the test results from [9] with evolved technology in operating systems, soundcard structures, and audio APIs, the research also evaluated some additional aspects which were not mentioned in earlier researches.

The general latency pictures of common operating systems were obtained. Though the lowest latency of an operating system with onboard soundcard can be close to the professional digital audio hardware, it may suffer losses of audio signals.

In additional to testing the effects on audio processing latency by CPU load outside audio applications, this research also measured whether latency is affected by the load coming from the audio processing application itself, especially with the large number of concurrent audio processing channels.

The latency of adaptive audio effects processing has also been evaluated. Due to the constraint of software structure, the value of the testing results is limited.

5.2. Future Study

Professional sound interface cards normally provide on-board DSP and mixing facilities for extremely low latency demanding applications. Although the onboard Intel HD audio architecture now has similar structure as the audio codec in professional audio devices, the signal path of common operating systems and onboard soundcards still suffers glitches, signal losses and distortions when the latency setting is considerable low.

Desktop computing has moved into the multi-core era whether adopting heterogeneous or homogenous architectures. With an integrated effect processor in the sound sub-system, it could be interesting to evaluate using the ability of the new parallel software structures to maintain the priority of the low latency audio processing path.

It would be challenging to satisfy both flexibility of emerging audio processing tasks and the stability of constant low latency in the audio signal path, especially when the flexibility of processing, routing, synchronising, and feature extraction over multiple channels is needed .

6. ACKNOWLEDGEMENTS

We would like to thank the researchers at the Centre For Digital Music, Queen Mary, University of London for advices on testing methods and considerations. We also would like to thank Birmingham Conservatoire and the Faculty of TEE of Birmingham City University for providing resources and equipments for testing.

7. REFERENCES

[1] Atau Tanaka, “Shifting Contexts for Computer Music, from Mainframes to DIY Culture”, Seminar presentation. Online abstract of the presentation: http://www.eca.ac.uk/cms_assets/Documents-Research/tanaka.pdf .

[2] M Guizani, “IEEE 802.20: Mobile Broadband Wireless Access”, IEEE Wireless Communications, 2007

[3] NP Lago, F Kon. “The Quest for Low Latency”, Proceedings of the International Computer Music Conference, November, 2004

[4] S. Farner, A. Solvang, A. Saebo, and U. P. Svensson, “Ensemble Hand-Clapping Experiments under the Influence of Delay and Various Acoustic Environments” Journal of the Audio Engineering Society, Vol 57 (12), pp. 1028-1041, December 2009

[5] Michael Lester, Jon Boley, “The Effects of Latency on Live Sound Monitoring”, Audio Engineering

Society Convention Paper 7198 Presented at the 123rd Convention 2007 October 5–8 New York, NY, USA

[6] N. Bouillot, et al., "AES White Paper: Best Practices in Network Audio," Journal of the Audio Engineering Society, Vol. 57 (9), pp. 729-741, September 2009

[7] Williston, Kenton; Bier, Jeff. “Trends and Directions in Signal-Processing Hardware for Audio Applications”. AES Conference: 23rd International Conference: Signal Processing in Audio Recording and Reproduction. May 2003

[8] Eli Brandt and Roger Dannenberg, “Low-latency music software using off-the-shelf Operating Systems”. Proceedings of the International Computer Music Conference.(ICMC-98), pp. 137-141. 1998

[9] K MacMillan, M Droettboom, I Fujinaga, “Audio latency measurements of desktop operating systems”, Proceedings of the International Computer Music Conference, September 2001

[10] Vincent Verfaille, Udo Zölzer, Daniel Arfib. “Adaptive Digital Audio Effects (A-DAFx): A New Class of Sound Transformations”. IEEE Transactions on Audio, Speech, and Language Processing, Vol. 14, No. 5, September 2006

[11] Intel, “High Definition Audio Specification, Revision 1.0”, April 15, 2004

[12] Realtek, “ALC885-GR, ALC885M-GR 7.1+2 Channel High-Performance HDA Codec with Content Protection Datasheet”. Rev. 1.1, 18 October 2006

[13] Portaudio, “an Open Source Cross Platform Audio API”. http://www.portaudio.com/

[14] JACK, “JACK Audio connection Kit”, online access http://jackaudio.org/files/docs/html/index.html

[15] Ableton, “Ableton Reference Manual version 8”, Live version 8.0.2 for Windows and Mac OS, May 2009

[16] John Tomarakos, “Interfacing the ADSP-21161 SIMD SHARC DSP to the AD1836 (24-bit/96kHz) Multichannel Codec”. Analog Devices, 2001