Audio Latency Measurements of Desktop Operating Systems

Karl MacMillan, Michael Droettboom, Ichiro Fujinaga

Peabody Institute of the Johns Hopkins University
email: {karlmac,mdboom,ich}@peabody.jhu.edu

Abstract

The realtime manipulation of audio with desktop computers has become both possible and popular over the last several years. Perhaps the most important aspect that determines the suitability of a computer operating system for this application is the latency of its audio system. This paper describes the results of an experiment measuring the audio latency of several current desktop operating systems, including Microsoft Windows, Apple MacOS, and Linux.

1 Introduction

With the dramatic increase in personal computer speed over the last several years, the possibilities for the realtime manipulation of audio have greatly expanded. Desktop computers are now sufficiently powerful to perform complex manipulations of audio in realtime. A responsive realtime audio system requires more than raw processing power, however; low audio latency is also necessary. Previous experiments that examined the latency performance of desktop operating systems revealed inadequate performance for many tasks (Brandt and Dannenberg 1998; Freed, Chaudhary, and Davila 1997). In this paper, the results of a new experiment to assess the status of low-latency audio manipulation with desktop operating systems are described. First, the definition of latency for the test is described. Then, the test method and the computer systems used in the test is detailed. Finally, the results of the experiment is presented.

2 Latency Definition

In the context of this paper, audio latency is defined as the minimum time required for a computer to store a sample from an audio interface into application memory and copy that same sample from application memory to the audio interface output. Both the conversions from analog to digital and back to analog are included. This definition deliberately includes the entire system in order to provide an indication of the actual performance of a computer system in a studio or stage setting. This means, however, that the results presented in this paper may show higher latencies than other measurements. We have seen many figures stated for the latency of various systems that reflect only sound output, buffer settings, or otherwise do not take into consideration the latency of the entire system. Though these are certainly valid measurements of parts of the audio systems, we felt that a measurement of the entire system would better show the latencies that can be expected from normal usage.

For this experiment, several types of applications, including some popular software samplers, are excluded. Specifically, applications written to run in kernel modules, separate realtime modules, or other non-standard methods are not considered. Applications written in this manner often sacrifice the benefits of running under general-purpose operating systems including support for protected memory, floating-point math, and standard soundcard drivers (Barabanov 1997; Van Buskirk and Bibbo 1999).

3 Test Method

The latency measurements were performed using a simple program that transferred samples to and from the soundcard using operating system-specific programming interfaces. In all tests, the audio was sampled at 44.1 kHz sampling rate with 16 or 24 bits of precision depending on the soundcard. The minimal latency was obtained through manual testing of a variety of buffer settings to determine the optimal settings for each system. The experimental setup consisted of a sound source (CD) connected directly to one channel of a DAT recorder and to the other channel through the computer. An impulse was recorded on the DAT with the left channel recording the direct sound and the right channel recording the same signal sent through the computer system. The latency was determined by counting the number of samples between the onset of the impulses on the left and right channels.

3.1 System Load

In addition to testing the systems with only the normal user interface and the latency test program running, the best performing system from each operating system family was tested while running two additional programs to generate unbounded CPU and disk load. The minimal latency was again obtained by manually testing buffer settings, but with the load generation programs running. These tests should give a general estimate of the degradation of latency

performance that can be expected while performing other complex tasks on the same computer, such as signal processing and hard disk streaming.

4   Test System Details

4.1   Hardware

The hardware used to test the system was comprised of recent model desktop and laptop computers. Several models of PCs and Macintoshes were used in the hopes of providing enough information to separate the hardware performance from that of the operating system. Also, wherever possible, the same hardware was used to test multiple operating systems. Table 1 details the general hardware used and Table 2 shows the details of the soundcards used.

System	CPU	Speed (MHz)	Memory (MB)	Disk	Soundcard
A	PIII	933	256	SCSI	1
B	G3	400	128	SCSI	2
C	G4	400	128	IDE	2
D	PIII	700	128	IDE	3
E	Celeron	366	192	IDE	4
F	PIII(dual)	933	512	SCSI	5
G	G4	500	256	IDE	6

Table 1. Hardware Details of the Test Systems.

Soundcard	Brand	Model	Grade
1	Crystal	CS4614	Consumer
2	Apple	Built-in	Consumer
3	ESS	Maestro 3	Consumer
4	ESS	Solo-1	Consumer
5	RME	Hammerfall (9652)	Professional
6	MOTU	2408	Professional

Table 2. Soundcard Details.

4.2   Operating Systems

In this experiment, eight separate operating systems were tested in order to give a comprehensive picture of the operating systems in current use. These were:

- Microsoft Windows 98
- Microsoft Windows 2000
- Microsoft Windows ME
- Apple MacOS 8
- Apple MacOS 9
- Apple MacOS X
- Linux 2.2
- Linux 2.4

The Linux systems all used RedHat 7.0 or RedHat 7.1. All of the operating systems were updated with the latest available patches.

With the Linux systems, several versions were tested in addition to the standard RedHat configuration. The kernel, which is the core of the operating system, was replaced with special versions designed to provide better latency. These

low-latency enhancements were Andrew Morton's "lowish-latency" patches, available from (www.uow.edu.au/~andrewm/linux/schedlat.html), and Ingo Molnar's low-latency patches (www.kernel.org/pub/linux/kernel/people/mingo/lowlatency-patches/). Though not part of the official Linux kernel, we thought it was important to test these enhancements because low-latency enhanced kernels are commonly added by users working with realtime audio and the DeMuDi Linux distribution (www.demudi.org), which is specifically targeted at multimedia users, includes them.

4.3   Programming Interfaces

One of the crucial aspects of obtaining the best latency performance is choosing the best application programming interface (API) for each operating system. Each of the operating systems tested presents several choices to the programmer. Table 3 lists common APIs and the platforms for which they are available. The APIs that are marked as standard are provided by default with the operating system.

API	Platforms	Standard
Microsoft DirectSound and DirectSoundCapture	Windows 98, ME, 2000	No
Microsoft Multimedia Extensions (MME)	Windows 98, ME, 2000	Yes
Steinberg Audio Stream Input Output (ASIO) 2.0	Windows 98, ME, 2000, MacOS 8 and 9	No
Apple SoundManager (SM)	MacOS 8, 9	Yes
Apple CoreAudio	MacOS X	Yes
Open Sound System (OSS)	Linux 2.2, 2.4	Yes
Advanced Linux Sound Architecture (ALSA)	Linux 2.2, 2.4	No

Table 3. Common Audio APIs.

In addition to those listed in Table 3, two other APIs were used that are designed to simplify audio programming. PortAudio (www.portaudio.com) provides a simplified API that is portable to Windows, Macintosh, and Linux. By providing a platform-neutral framework, PortAudio allows programmers to create audio applications that are source-code compatible on many systems. LAAGA (Linux Audio Application Glue API),   an experimental system designed by Paul Davis (www.op.net/~pbd) allows the transparent sharing of audio data between applications on Linux. The system, which does not need to run in the kernel and uses only standard Unix programming interfaces, was not originally part of the experiments for this paper, but the results were impressive enough to warrant inclusion.

5   Results

Table 4 summarizes the results of this experiment for unloaded systems. The system labeled 'Spirit' refers to a Spirit Digital 328 Mixer by Soundcraft, which was included to provide a baseline comparison. The Linux systems are labeled by kernel version. Only one test was done under Linux using the OSS (Open Sound System)

(www.opensound.com/oss.html) because of a lack of reliable support for full duplex with the test system hardware. Table 5 shows the results of the tests with system load.

System	Operating System	Details	Latency (ms)
Spirit	-	-	1.81
A	Linux 2.4.1 (AM)	ALSA	2.72
A	Linux 2.4.1	ALSA	2.72
F	Windows 2000	ASIO	3.11
C	MacOS X	CoreAudio	3.97
F	Linux 2.4.1 (IM)	ALSA (L)	4.30
F	Linux 2.4.5 (AM)	ALSA (L)	4.30
F	Linux 2.4.2	ALSA (L)	4.30
G	MacOS 9.04	ASIO	6.80
A	Linux 2.2.16	ALSA	7.19
F	Windows 2000	MME (P)	11.45
E	Linux 2.4.0	OSS	12.20
E	Windows 98	MME (P)	60.86
E	Windows 98	DirectSound (P)	63.24
D	Windows ME	MME (P)	73.51
A	Windows 2000	MME (P)	75.85
D	Windows ME	DirectSound (P)	82.86
B	MacOS 8.6	SM (P)	106.24
A	Windows 2000	DirectSound (P)	123.11
C	MacOS 9.04	SM (P) VM Off	195.58
C	MacOS 9.04	SM (P) VM On	935.53

Table 4. Latency Test results for systems without load. P – PortAudio. L – LAAGA. VM – MacOS virtual memory settings. AM – Andrew Morton. IM – Ingo Molnar.

System	Operating System	Details	Latency (ms)
C	MacOS X	CoreAudio	3.97
F	Linux 2.4.1 (IM)	ALSA (L)	4.30
F	Linux 2.4.5 (AM)	ALSA (L)	4.30
F	Linux 2.4.2	ALSA (L)	4.30
F	Windows 2000	ASIO	6.03
G	MacOS 9.04	ASIO	6.80
F	Windows 2000	MME (P)	245.17

Table 5 – Latency Test Results with system load. P – PortAudio. L – LAAGA. AM – Andrew Morton. IM – Ingo Molnar.

5.1   Linux

The Linux systems had the best performance on the tests without load and the second best performance with load. Furthermore, the Linux systems tended to have the least difference between best-case and worst-case performance. Excluding the test that used the less efficient OSS API, which is slated for replacement by ALSA (Advanced Linux Sound Architecture) (www.alsa-project.org) in the next kernel version, all of the Linux tests had latencies below 10 milliseconds. Finally, the performance of the Linux systems are independent of the quality and expense of the soundcards.

The tests using the kernels with low-latency enhancements did not show significantly better performance than the standard kernels. This is most likely due to the test hardware used and the simplicity of the load generation programs. The low-latency patches address issues only with specific kernel sub-systems rather than providing a general enhancement to the latency characteristics of the system. It is likely that the load generation programs used did not exercise the improved sub-systems. Additionally, in our experience, most of the latency problems in Linux are related to IDE disks, which none of the systems tested with load used. Performing other tasks with the systems while running the latency test program, such as CPU intensive graphics programs and compiling large projects, showed that some disk-intensive tasks can cause problems with the standard kernel. No amount of CPU load alone created latency problems, however.

One interesting aspect of these results is the excellent latency performance using LAAGA. The only other results that were less than 10 milliseconds were ASIO (Audio Streaming Input Output) (Steinberg 1999) on Windows and Macintosh, neither of which allow more than one application access to the sound hardware concurrently as LAAGA allows. LAAGA showed excellent performance that did not noticeably degrade with multiple test applications accessing the sound hardware concurrently.

5.2   Microsoft Windows

The Windows systems tested had results ranging from excellent to poor in correlation with the quality of the soundcard being tested. All of the tests using the RME Hammerfall card without system load performed well, but all of the tests with consumer-quality soundcards had latencies above 60 milliseconds. For the tests with system load, only the ASIO drivers performed well. The test program using the MME (Multimedia Extensions) (Petzold 1998) driver performed poorly with load. Preliminary tests showed that this was caused more by CPU load than disk load.

5.3   Apple MacOS 8 and 9

Like the Windows systems, the classic MacOS systems had both excellent and poor performance, again correlated with soundcard quality. The performance with ASIO, which was tested using Max/MSP, was excellent in both the loaded and unloaded tests. One surprising result was the latency test using MacOS 9.04 with virtual memory enabled, which yielded latencies of nearly 1 second.

5.4   Apple MacOS X

Apple MacOS X was the second best performing system on the tests without load and the best performing system on the tests with load. The new CoreAudio API, which was introduced with MacOS X (Apple 2001), is clearly designed

for low-latency performance. In addition, CoreAudio allows multiple programs to access the soundcard concurrently. Only LAAGA on Linux provided comparable performance to MacOS X while allowing multiple programs access to the soundcard.

6   Conclusions

All of the current desktop operating systems offer excellent latency performance under some conditions, though most of them cannot deliver this performance in all situations. This is a substantial improvement over previous results (Brandt and Dannenberg 1998; Freed, Chaudhary, and Davila 1997), but because of the inconsistency of the results more improvement is necessary before reliable low-latency performance can be expected from desktop operating systems.

In conclusion, Linux showed the best performance in the tests without load while MacOS X showed the best performance in the tests with load. Windows and MacOS 8 and 9 produced some of the best results when using a professional soundcard with the ASIO API but showed poor performance when using the standard APIs and consumer-grade soundcards.

7   Acknowledgements

References

Apple. 2001. *Audio and MIDI on Mac OS X: Preliminary Documentation.* Apple Computer, Inc.

Barabanov, M. 1997. A Linux-based real-time operating system. MS thesis, New Mexico Institute of Mining and Technology.

Van Buskirk, J. E., and J. A. Bibbo. 1999. Filed June 10, 1998. Synthesizer system utilizing mass storage devices for real time, low latency access of musical instrument digital samples. United States Patent No. US6,008,446.

Brandt, E., and R. B. Dannenberg. 1998. Low-latency music software using off-the-shelf operating systems. In *Proceedings of the International Computer Music Conference,* 137–41.

Freed, A., A. Chaudhary, and B. Davila. 1997. Operating systems latency measurement and analysis for sound synthesis and processing applications. In *Proceedings of the International Computer Music Conference,* 479–81.

Petzold, C. 1998. *Programming Windows.* 5th ed. Redmond, WA: Microsoft Press.

Steinberg. 1999. *Steinberg Audio Streaming Input Output Specification: Development Kit 2.0.* Steinberg Soft- und Hardware GmbH.