

ASIO Server

Pedro Sousa, Tiago Nogueira, Nuno Fonseca

ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO DE LEIRIA

Morro do Lena – Alto do Vieiro, 2401-951 Leiria, Apart. 3063

Tel. +351 244820300, Fax. +351 244820310, <http://www.estg.ipleiria.pt>

Email: eic13482@student.estg.ipleiria.pt, eic13475@student.estg.ipleiria.pt, nfonseca@estg.ipleiria.pt

1. Resumo

Este projecto consistiu na implementação de um servidor/cliente de placas de som ASIO. Num sistema convencional o som apenas podia ser tratado num único computador. Através do ASIO Server é possível distribuir o processamento por vários computadores que comunicam através da rede local, um desses computadores funciona como servidor e é o único que possui uma placa de som compatível com a tecnologia ASIO. Os outros computadores funcionam como clientes, acedendo aos canais da placa de som do servidor através de um driver virtual. Foram efectuados testes de performance às aplicações implementadas.

2. Introdução

2.1 Tecnologia ASIO

Uma placa de som pode executar várias tarefas simultaneamente, como por exemplo gravar som proveniente de uma ou várias entradas da placa e reproduzir áudio de várias fontes distintas. A velocidade de processamento e o sincronismo desempenham um papel fundamental.

O subsistema áudio do sistema operativo Windows XP da Microsoft possui um valor elevado de latência, entre 25 a 50 milissegundos, este tempo corresponde ao período que os dados demoram desde a aplicação de *software* até ao *hardware* da placa de som.

De forma a melhorar a latência uma empresa alemã Steinberg desenvolveu uma tecnologia denominada ASIO (Audio Stream In/Out). Esta tecnologia permite obter baixos valores de latência durante a manipulação de áudio digital. A tecnologia ASIO permite contornar os sistemas operativos, como Windows ou Mac, criando assim uma comunicação mais eficiente entre a placa de som e o *software* áudio, pois interage directamente com o *hardware* áudio. Esta tecnologia permite também que o *software* tenha acesso a vários canais, contrariamente aos convencionais dois canais que o sistema áudio do Windows XP disponibiliza.

2.2 Sistema ASIO Convencional

Ao utilizar o *driver* ASIO em conjunto com uma placa de som que o suporte, é possível tratar independentemente os diferentes canais, sejam de *input* ou de *output*, de som da placa.

O seguinte diagrama mostra o funcionamento de um sistema ASIO convencional, constituído por um *driver* ASIO e uma aplicação de tratamento áudio. Neste sistema não intervêm as aplicações implementadas neste projecto, o sistema é constituído apenas por uma máquina.

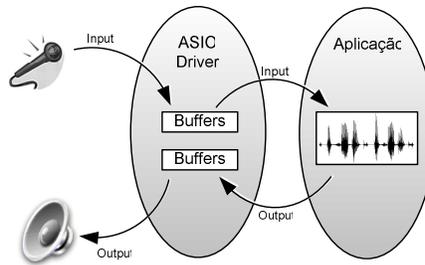


Figura 1 - Sistema ASIO convencional

O *driver* ASIO interage com a placa de som e disponibiliza à aplicação, através de *buffers*, o áudio de *input*. Do mesmo modo disponibiliza *buffers* de *output* para a aplicação preencher com o áudio que pretende enviar para os canais de *output*. Todo o funcionamento do sistema é síncrono e controlado pela placa de som.

2.3 Sistema ASIO Server

O diagrama seguinte mostra um sistema semelhante ao anterior, mas que tira partido das aplicações implementadas neste projecto (ASIO Server e ASIO Client Driver). A aplicação ASIO Server é vista pelo *driver* ASIO como uma aplicação de tratamento de som comum e vai comunicar com o ASIO Client Driver que por sua vez, do ponto de vista da aplicação de tratamento áudio, faz o papel de um *driver* ASIO normal. Note-se que o processo de transferência dos *buffers* de áudio pela rede vai introduzir latência no sistema, o que pode fazer com que os *buffers* de *output* não cheguem assim que a placa de som requer. Assim, para além dos *buffers* existentes nos *drivers* (ASIO Driver e ASIO Client Driver) será também necessário recorrer a *queue buffers* no servidor (ASIO Server).

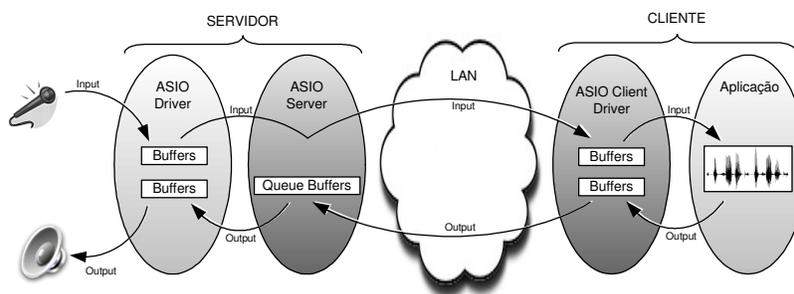


Figura 22 – Diagrama ASIO com recurso ao ASIO Server

ASIO Server – É a aplicação servidora. Permite seleccionar um *driver* ASIO existente no servidor e partilhar os canais áudio existentes na placa de som ASIO.

ASIO Client Driver – É uma aplicação constituída por um DLL e um painel de configuração. O DLL efectua uma ligação ao servidor permitindo a transferência de *buffers* de áudio. O painel de configuração vai permitir, de entre outras configurações seleccionar, dos canais disponíveis no servidor, os que se pretendem utilizar.

3. Testes

Pretende-se com este testes verificar o impacto da latência introduzida pela rede no desempenho do sistema. Os testes foram efectuados numa rede a 100Mbps.

O *Round Trip Time (RTT)* é considerado como o tempo que passou desde que o pedido de *buffers* de *output* foi enviado pela rede para ao cliente até ao instante em que os *buffers* de *output* do mesmo cliente chegaram ao servidor. O *RTT* vai ser influenciado não só pelo desempenho da rede, bem como pelo desempenho de todo o sistema, incluindo a implementação e optimização da aplicação servidora e cliente.

A latência de *output* é o tempo que passou desde que um *buffer* de áudio foi disponibilizado pela placa de som, até ao instante em que o mesmo *buffer* foi transmitido de volta para a placa de som já processado pela aplicação que o tratou, por outras palavras é a ‘idade’ do *buffer* de *output*. Este tempo é constante e os *buffers* de *output* devem estar todos disponíveis após o mesmo, de modo que o *RTT* nunca pode ser superior à latência de *output*.

3.1 Latência introduzida pela rede (um cliente)

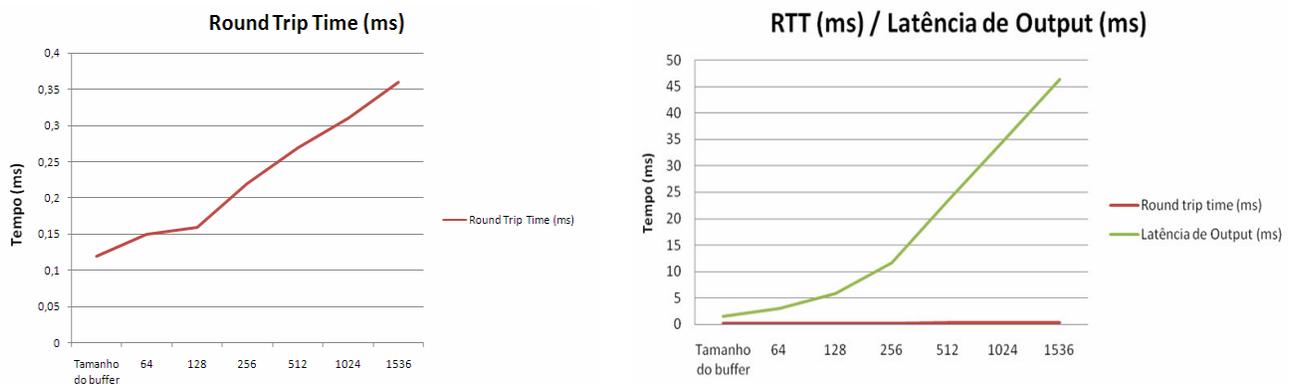


Figura.3 Round Trip Time e Latência de Output em função do tamanho do *buffer*

Os resultados anteriores foram obtidos apenas com um cliente ligado ao servidor e com pouco tráfego na rede, pelo que foi possível obter resultados excelentes para os valores de latência. Como se pode observar o tempo que os *buffers* demoram a ser pedidos e enviados para o servidor nunca ultrapassa a latência de *output*, ou seja, nunca houve um *buffer* de *output* a chegar atrasado. É também possível observar que não compensa ter tamanhos de *buffer* muito elevados, pois o tempo de latência de *output* aumenta bastante e o *RTT*, para tamanhos de *buffer*

menores, é suficientemente baixo para fazer chegar os *buffers* de áudio a tempo. Assim, é preferível manter um tamanho de *buffer* de aproximadamente 512 amostras, para precaver possíveis atrasos, obtendo assim uma latência satisfatória.

3.2 Latência introduzida pela rede (vários clientes)

No gráfico seguinte está representado o valor de *Round Trip Time (RTT)* obtido com vários clientes ligados ao servidor. O teste foi efectuado com um tamanho de *buffer* áudio de 512 amostras e uma taxa de amostragem de 44100Hz, o que resulta num tempo de aproximadamente 12ms entre a troca de *buffers* de áudio. O *RTT* indicado é o máximo que foi obtido ao analisar o valor de todos os clientes.

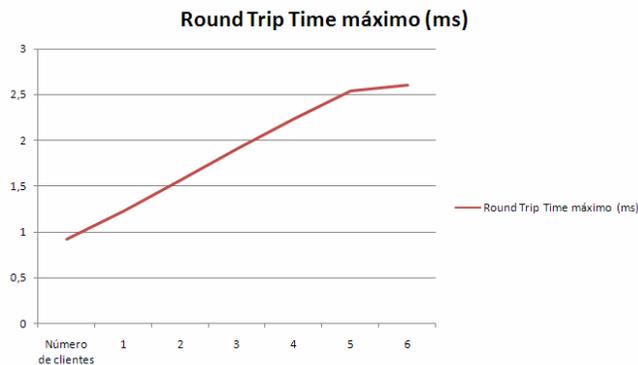


Figura.4 Round Trip Time / Número de clientes

Pode-se observar que a o *RTT* varia linearmente com o número de clientes, podendo assim ser estimado um número máximo de clientes de acordo com a latência que se deseja obter. Este teste decorreu durante 20 minutos, ao longo dos quais foram progressivamente adicionados clientes e durante os quais não se perdeu nenhum pacote. Durante a fase de teste com 5 clientes, houve um *buffer* de *output* que chegou atrasado, ou seja, não chegou a tempo de ser enviado para a placa de som assim que esta requereu, pelo que, numa situação prática seria necessário aumentar o tamanho dos *buffers* de áudio, e conseqüentemente o tempo entre leituras de *buffers* por parte da placa de som, ou então, e talvez mais sensato, aumentar o *queue buffer* apenas do cliente cujo pacote se atrasou.

4. Conclusão

Neste projecto foram implementados um servidor ASIO e o respectivo cliente. Através do *driver* virtual implementado, foi possível obter valores latência iguais aos obtidos utilizando a placa de som na própria máquina, tornando assim possível distribuir processamento áudio por vários computadores numa rede garantindo um valor mínimo de latência.

5. Bibliografia

[1] ASIO SDK - Steinberg Audio Streaming Input Output Specification, Development Kit 2.2