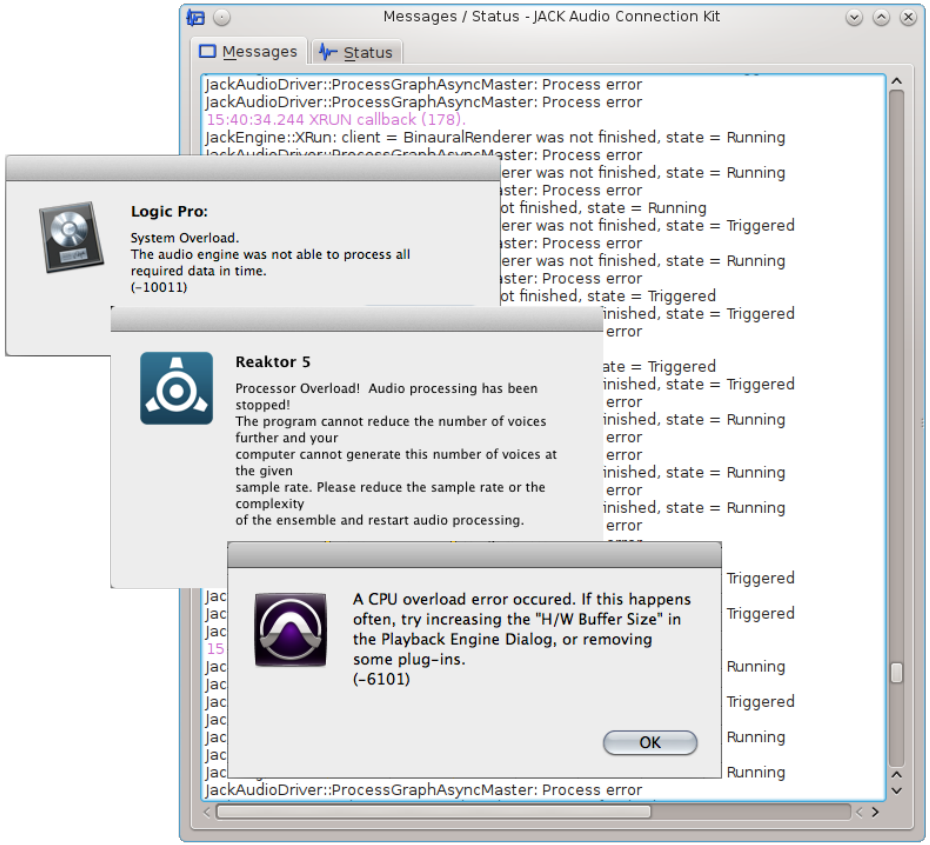
Projeto de Pesquisa: Processamento de áudio com custos computacionais flexíveis

Thilo Koch
Grupo de Pesquisas em Computação Musical
IME - USP

18/11/2015

Problematização

- Problemas de sobrecarga acontecem em muitos cenários.
- Limite de capacidade de computação é inerente à ideia de hardware.
- Upgrade nwm sempre é desejável ou viável.



Interesse em pesquisar sistemas que administrem melhor os recursos.

Problemas relacionados

- Sistemas de tempo real:
  - garantem tempo de entrega
  - encontrados na área de controle de sistemas
  - computação flexível / computação imprecisa
- Redes
  - Qualidade de serviço (QOS): descrição de serviços e mecanismos de controle
  - *Datagram Congestion Control Protocol* (DCCP)
- Compressão de dados
  - *trade-off* entre tempo gasto com compressão / descompressão e espaço na memória
  - ou tempo de transmissão em uma rede
- *trade-off*: milhares de exemplos: memória cache, ... ,*fast food*

Proposta

- Desenvolver uma metodologia que permita a realização de um *trade-off* flexível entre custos computacionais e qualidade do resultado sob a condição de tempo real para aplicações interativas.
- Como: parametrizar os elementos de processamento e controlar carga computacional.
- Objetivo: menor perda de qualidade possível do ponto de vista da experiência do usuário.

Questões de pesquisa 1

- Como um sistema poderia realizar o *trade-off* flexível entre custos computacionais e a qualidade percebida do resultado do processamento?
- Como poderíamos quantificar a perda de qualidade de experiência do resultado auditivo em relação aos parâmetros que controlam o processamento?
- Como realizar um método de gerenciamento do sistema para minimizar a perda de qualidade adaptando dinamicamente os custos computacionais aos recursos disponíveis?

Questões de pesquisa 2

- Um sistema desse tipo seria capaz de resolver ou pelo menos aliviar o problema de sobrecarga? Em quais casos?
- Quais são as aplicações onde a nossa abordagem é mais eficaz?
- Seria possível propôr um procedimento generalizado para aplicar nossa solução a outros sistemas?

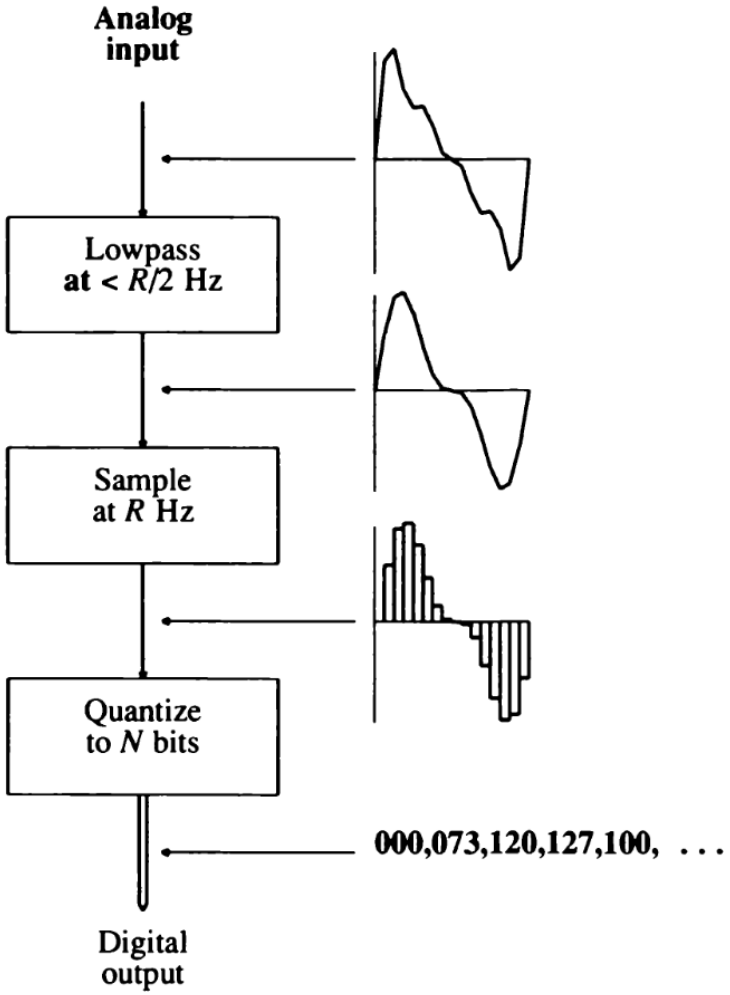
Metodologia

- Avaliar a correspondência entre a flexibilização dos custos computacionais e os efeitos sobre a qualidade de experiência.
- Critérios de avaliação por revisão bibliográfica e por entrevistas com usuários.
- Análise e parametrização de elementos de processamento; desenvolver métodos de controle.
- Avaliação experimental para configurações específicas para avaliar a metodologia desenvolvida.

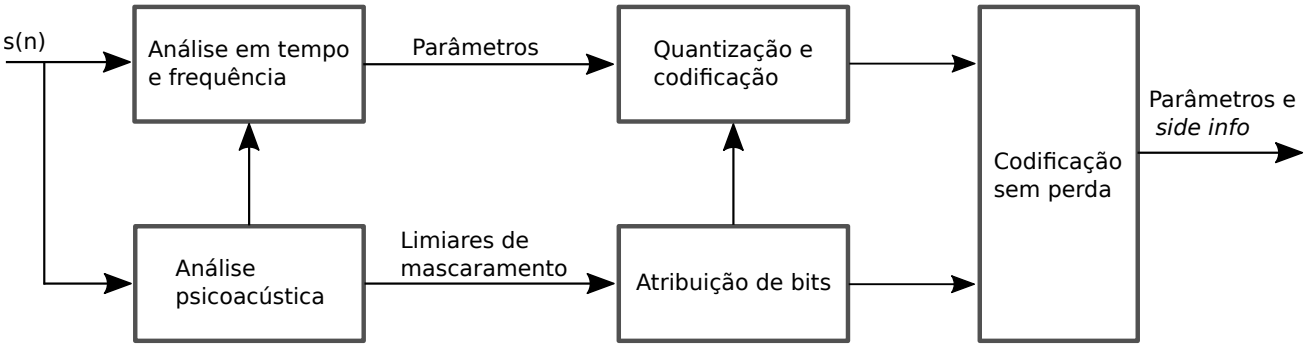
Representação e codificação de áudio

Aspectos do processo de conversão de ondas sonoras em dados digitais de áudio

- Discretização e frequência de Nyquist
- Quantização e ruído de quantização
- Vários tipos de codificação sem perda de qualidade
- Existe representação no domínio da frequência equivalente



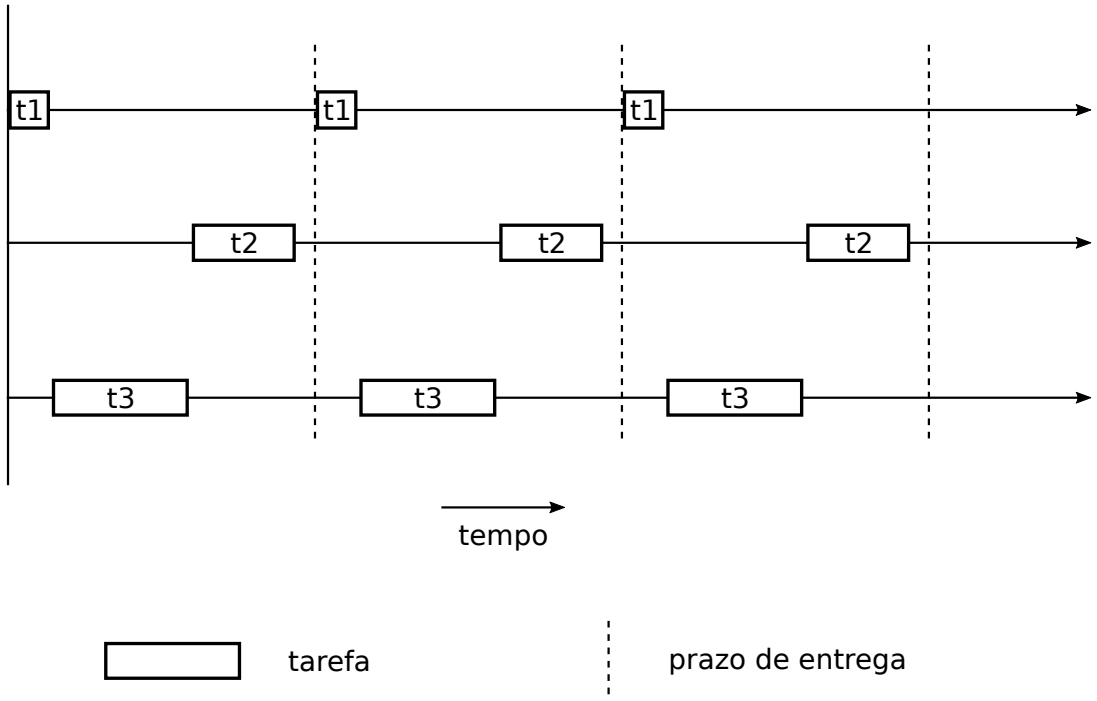
Codificação perceptual de áudio



- Limiar absoluto de audibilidade
- Bandas críticas e mascaramento
- Mascaramento simultâneo e temporal
- Entropia perceptual
- *trade-off* entre taxas de compressão e qualidades diferentes

Tempo real

- Sistema que garante os tempos de entrega de tarefas
- Sistemas de áudio podem ser descritos com modelo periódico
- Tarefas correspondem a módulos do processamento de áudio
- Sistemas como Linux, Windows NT ou OS X não são tempo real *stricto sensu*
- Estratégia: priorizar *RT threads* e esperar que tudo dê certo



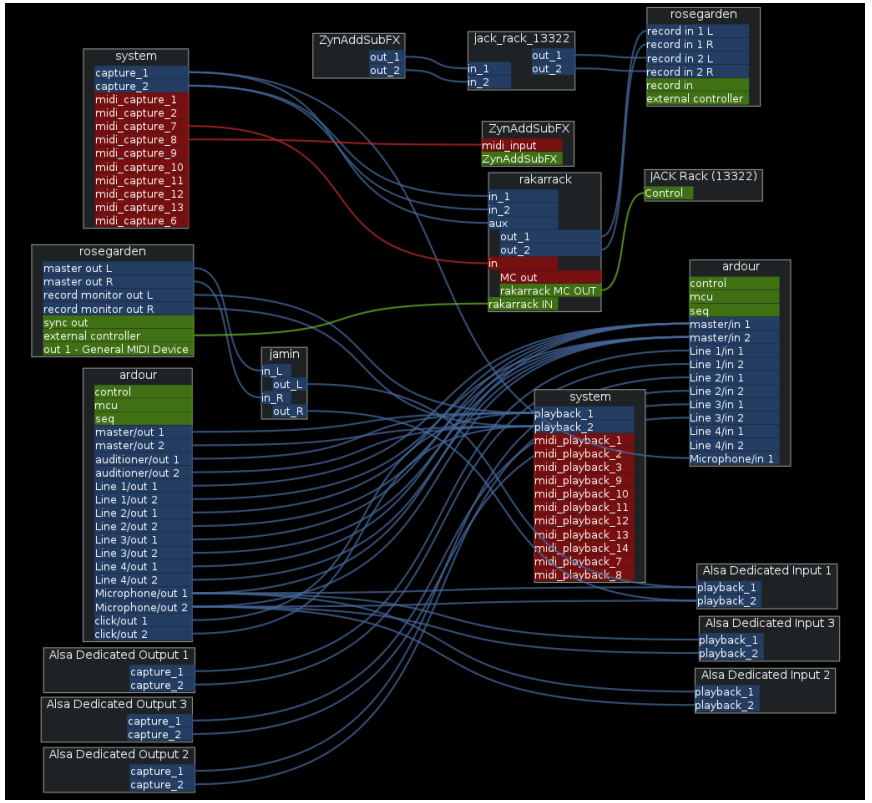
Computação flexível

- Conceito introduzido para realizar *trade-off*.
- Observação: em certos casos, um resultado com uma qualidade menor no momento devido é melhor que o resultado exato porém atrasado.
- Tarefas são divididas em componentes obrigatórias e componentes opcionais.
- Vários métodos, por exemplo:
  - Peneira (*sieve*),
  - *Milestone*,
  - Várias versões.

Ambientes para processamento de áudio 1

Jack Audio Connection Kit

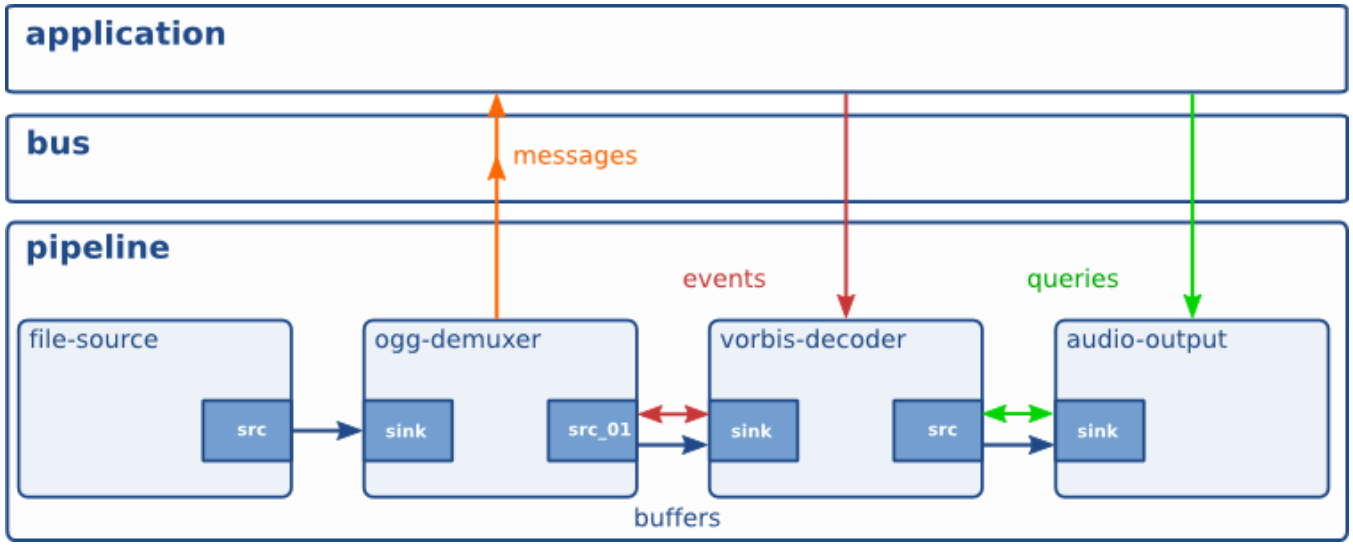
- modelo cliente servidor
- conectar componentes
- baixa latência, baixa complexidade
- muito difundido em Linux
- existe para muitos sistemas POSIX



Ambientes para processamento de áudio 2

GStreamer

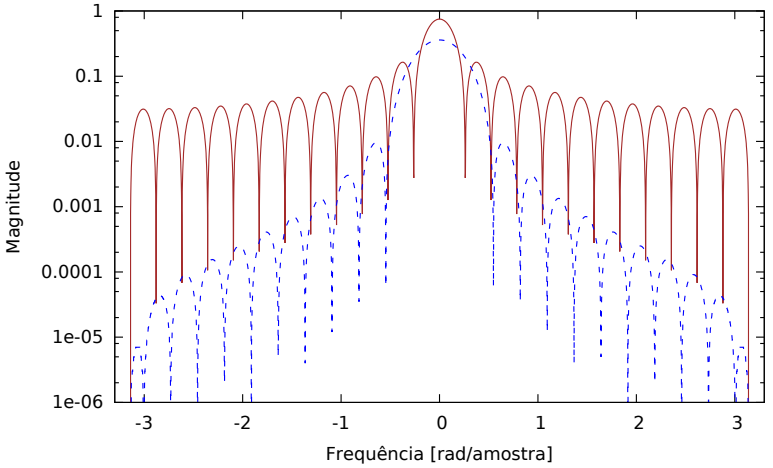
- Arcabouço para implementar programas e *plugins*
- Estrutura das aplicações: *pipelines* de elementos de processamento.
- Camada de comunicação complexa



Filtros FIR

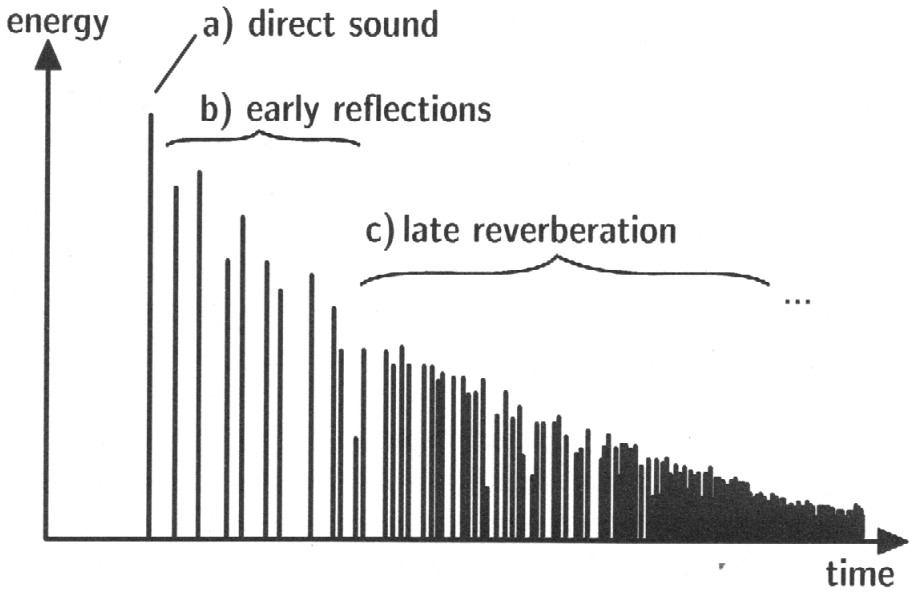
- Definição: $y_n = \sum_{i=0}^{M-1} a_i x_{n-i}$ (a_i coeficientes do filtro)
- aplicação no domínio do tempo: convolução
  - custos $O(NM)$
- no domínio da frequência: convolução circular com *overlapp-add*
  - custos $O\left(N \frac{M+L}{L} log(M+L)\right)$

- **parâmetro**: comprimento do filtro (M)
- "encurtar"entendido como janelamento $\rightarrow$ funções sinc



Simulação de salas

- Resposta impulsiva (RI)



- Modelos de salas, **parâmetro**: detalhes
- Modelo de fontes virtuais, **parâmetro**: número de reflexões
- Traçado estocástico de raios, **parâmetro**: número de partículas

Espacialização

Síntese do campo sonoro

- Ressintese de um campo sonoro expandido, por meio da interferência de sinais.
- $Y_m(x,\omega) = \sum_{n=1}^{P} (S_{FP}\ F(\omega)\ A(x)\ D(x,\omega))$
- Não usar todas as fontes virtuais para todo os alto-falantes

Síntese binaural

- Sintetizar um sinal a ser reproduzido diretamente no ouvido de um ouvinte.
- $y(t) = \sum_{i=1}^{P} s_i(t) * RI_i(t)$
- *Head related transfer function* (HRTF) dependem do ângulo de incidência
- Flexibilização: versões dos HRTFs simplificadas, resolução angular

Cadeias e grafos de processamento

- Maior conhecimento do fluxo de dados (análise de grafos) $\rightarrow$ outras opções de parametrização
- Taxa de amostragem, quantização: conhecimento do conteúdo (espectral)
- *Pipeline* no domínio da frequência:
  - Aplicar critérios psicoacústicos para reduzir números de valores na representação no domínio da frequência,
  - computar com menos valores no processamento subsequente.
- Número de fontes virtuais: limiar de audibilidade e *clustering*

Controle e gerenciamento de recursos

- Controlar: qualidade máxima com recursos computacionais disponíveis
- A partir das relações: parâmetros - qualidades.
- Tempo disponível: $\Phi(n) \geq \sum_{i=1}^{N} \phi_i(n)$
- Erro: $E_i = f_i(\phi_i, E_{i-1})$
- Como controlar eficientemente e corretamente o processo de processamento de áudio por inteiro?

Limitações

Aplicabilidade

- Parâmetros dos algoritmos fixos

Exequibilidade da implementação

- Arcabouço pouco flexível

Limites de qualidade

- Perda de qualidade inadmissível

Implementação prova de conceito

- Implementação como campo de experimentação
- Estratégia: reuso máximo de código (invasão mínima) para melhor difusão
- Usar arcabouços e software existentes
- "Sobrepôr"escalonamento (escolher modelo)
- Revisar opção de *pipelines* no domínio da frequência

Perguntas?

Obrigado!