Transcodificação Distribuída de Vídeo de Alto Desempenho

João Bernardo Vianna
Laboratório de Computação Paralela
e Sistemas Móveis
Universidade Federal do Rio de Janeiro
jb.vianna@ufrj.br

Diego L. C. Dutra
Laboratório de Computação Paralela
e Sistemas Móveis
Universidade Federal do Rio de Janeiro
diegodutra@compasso.ufrj.br

Lauro Luis A. Whately
Laboratório de Computação Paralela
e Sistemas Móveis
Universidade Federal do Rio de Janeiro
whately@compasso.ufrj.br

Claudio L. de Amorim
Laboratório de Computação Paralela
e Sistemas Móveis
Universidade Federal do Rio de Janeiro
amorim@compasso.ufrj.br

ABSTRACT
This paper proposes and evaluates an efficient distributed video transcoding. Unlike known solutions that require modification of the encoder or restrict the performance of the solution, we present a optimized technique for video search, allowing the creation of independent segments that can be transcoded in parallel. Preliminary experiments demonstrate that the distributed transcoding in a cluster of 4 nodes runs 5.6 times faster in comparison with the same video transcoding running 48 threads on a dual quad-core node.

Categories and Subject Descriptors
H.4 [**Information Systems Applications**]: Miscellaneous; D.2.8 [**Software Engineering**]: Metrics—*performance measures*

General Terms
Experimentation

Keywords
video, transcoding, distributed computing

1. INTRODUÇÃO
A proliferação de dispositivos móveis, como *smartphones* e *tablets*, e também *webTVs*, aumentou muito a demanda por vídeo na internet, nos últimos anos. Consequentemente, a codificação se torna uma operação importante, para que os vídeos estejam no formato, qualidade e taxas compatíveis com os respectivos dispositivos. A codificação de vídeo é uma tarefa de alta complexidade computacional e intensivo uso do processador, o que significa que o processamento pode levar durar muito tempo. Por exemplo, um vídeo com taxa de 1.5Mbps de 2 horas pode levar até 3 horas para transcodificar em um servidor quad-core.

Encontrar um modo de distribuir o processamento entre vários servidores reduziria o tempo de codificação e consequentemente, tornaria mais rápida a geração dos vídeos para os dipositivos desejados. Com esse objetivo, desenvolvemos um método para particionar o vídeo em vários segmentos e distribuir a transcodificação de cada segmento entre vários servidores de um *cluster*, resultando em uma transcodificação mais rápida quando comparada com um servidor apenas. Implementando o método em um sistema de transcodificação distribuída obtemos uma solução genérica e custo-efetiva em software que apresenta alto desempenho na geração de vídeos em diferentes taxas.

O desafio na segmentação é como dividir o vídeo original de forma que depois da transcodificação distribuída, a reintegração dos segmentos e a sincronização do áudio e vídeo possam ser feitas de modo correto. O sistema separa as transcodificações de áudio e vídeo. Enquanto o vídeo, que demanda maior tempo de processamento, é dividido em segmentos e distribuídos por vários servidores, o áudio é codificado integralmente em um único servidor.

Este artigo é organizado da seguinte forma: na seção 2 apresentamos os trabalhos relacionados; na seção 3 descrevemos o método de segmentação baseada em busca e a implementação do sistema de transcodificação distribuída; discutimos os experimentos e respectivos resultados na seção 4 e por fim, as conclusões estão na seção 5.

2. TRABALHOS RELACIONADOS
Os trabalhos relacionados com essa proposta apresentam soluções que manipulam o codificador ou adotam compromissos que restrigem o desempenho da solução.

Propostas que implementam a transcodificação distribuída através do codificador restrigem os vídeos que podem ser usados à uma tecnologia específica de codificação, como em CloudStream [3]. As soluções que não trabalham com compressão temporal, também exibem o mesmo tipo de restrição em relação à codificação dos vídeos, encontrado em

Split&Merge [5]. Outras propostas, por exemplo o trabalho de [7], restrigem o desempenho da solução quando implementam a decodificação para retirar a compressão temporal antes da codificação distribuída dos segmentos do vídeo.

3. TRANSCODIFICAÇÃO DE VÍDEO DISTRIBUÍDA

A codificação de um vídeo com compressão temporal (como MPEG2, H.264/AVC e VP6, entre outros) utiliza diferentes tipos de quadros. Na especificação do MPEG4 [6], por exemplo, existem os quadros I, P e B, dos quais o primeiro é um quadro completo e os outros dois utilizam referências de quadros próximos para formar uma imagem completa. Em um segmento contendo apenas GOP-fechado, todos os quadros podem ser decodificados independentemente. Ao contrário, se o primeiro GOP de um segmento for um GOP-aberto, o GOP anterior é necessário para decodificar quadros do tipo B no segmento. Por esta razão, um vídeo H.264 não pode ter um segmento transcodificado (uma decodificação, seguida de codificação em outra taxa ou formato) de forma independente se não possuir todas as referências necessária aos quadros B contidos nele.

As soluções de transcodificação distribuída existentes modificam o codificador para garantir que as referências de quadros P e B estejam contidas em grupos de quadros bem definidos ou realizam a decodificação completa do video - para retirar a compressão temporal - e só após segmentam o vídeo para distribuir a codificação. No sistema de transcodificação aqui proposto, um arquivo de vídeo pode ser segmentado em quantas divisões forem necessárias e estas são distribuídas entre nós remotos para a transcodificação. Mas a segmentação não divide o vídeo em arquivos menores, apenas cria os parâmetros necessários para gerar os segmentos. Todos os nós remotos possuem o vídeo original completo, assim o processo de transcodificação tem acesso às referências necessárias ao decodificar cada segmento. Assim, evitamos a perda da informação ao dividir o vídeo em segmentos. No processo de codificação, de cada segmento é gerado um novo arquivo. Ao final, no nó mestre, estes arquivos são multiplexados com o áudio para gerar o novo vídeo. A segmentação e a correspondente integração dos segmentos devem ser simples e rápidas para que não aumentem desnecessariamente o tempo total de transcodificação do vídeo.

A faixa de áudio não pode ser segmentada. O áudio por conter dados entre quadros consecutivos de vídeo, apresentam artefatos quando segmentado, ocasionando perda de informação. Assim, a transcodificação do áudio se dá sequencialmente e é independente da codificação do vídeo. A transcodificação sequencial de áudio pode ser um fator limitante para o desempenho alcançado na transcodificação do vídeo.

3.1 Distribuição de tarefas

A segmentação é iniciada no nó mestre, com a análise dos metadados do vídeo. Nesta etapa é selecionado os respectivos pontos de segmentação e duração do segmento, de acordo com a duração, taxa de bits e quantidade de frames do vídeo. Após receber os parâmetros da segmentação, cada nó de transcodificação inicia o processamento dos respectivos segmentos (veja Figura 1). O nó mestre possui uma área do sistema de arquivo compartilhada com os nós de transcodificação, onde pode ser encontrado o vídeo original.

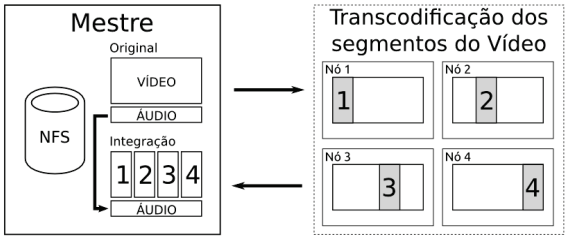

Figura 1: Etapas da transcodificação distribuída com um servidor mestre e 4 servidores de transcodificação

Cada nó executa a transcodificação simultânea de todos os segmentos recebidos, gerando arquivos no formato desejado sem um *container*. Ao final, o nó mestre pode encontrar, na área compartilhada, todos os arquivos gerados pelos nós de transcodificação e assim concatena-los em um só arquivo, sicroniza-lo com o áudio codificado e inseri-lo em um *container*.

3.2 Segmentação de Vídeo por Busca Otimizada

A nossa solução para segmentar o vídeo sem perder as referências entre os quadros, requer que os nós de transcodificação acessem o arquivo do vídeo original e caminhem por ele até alcançar o ponto de início da conversão. Convencionalmente, uma ação de busca num arquivo de vídeo faria a decodificação quadro a quadro até o ponto especificado. Desta forma, cada um dos N processos realiza a decodificação de todo o vídeo anterior ao seu trecho de trabalho. O que equivale ao processo N decodificar todo o vídeo, anulando o ganho de desempenho na distribuição da transcodificação.

A nossa técnica de busca foi aprimorada para reduzir o percentual de decodificação extra necessário na definição do segmento. A idéia é fazer uma busca indexada pelo quadro I e procurar qual está mais próximo do início do segmento e a partir daí, decodificar os quadros até o fim do segmento. A Figura 2 ilustra a escolha dos segmentos e o posicionamento do quadro I que será usado na decodificação do respectivo segmento. O tempo de busca do quadro I é praticamente desprezível comparado com a decodificação quadro-a-quadro na busca convencional.

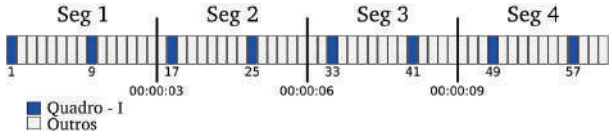

Figura 2: Segmentação do arquivo de vídeo.

A Figura 2 mostra um vídeo com taxa de 5 quadros por segundo e um total de 60 quadros. Para 4 servidores de transcodificação, o sistema seleciona os tempos 0, 3, 6 e 9 segundos como os devidos limites e atribui 15 quadros a cada segmento. Através da mesma Figura, podemos descrever a criação do segmento 2. O segmento foi delimitado pelos tempos 3 e 6 segundos. Na transcodificação é feita uma

busca do quadro I mais próximo e anterior ao segmento. Encontrado o quadro 9, este é usado como referência para a decodificação do segmento 2.

4. AVALIAÇÃO EXPERIMENTAL E RESULTADOS

Para avaliar o sistema de transcodificação distribuído proposto realizamos experimentos em um cluster composto de 5 nós: um nó mestre, responsável pela segmentação, escalonamento de tarefas e a integração final dos segmentos transcodificados; e mais 4 nós, responsáveis pela transcodificação dos segmentos.

Tabela 1: Ambiente Experimental - Nó Computacional

Processadores	2 x Intel Xeon E5620 4-Core HT
Memória	16GB ECC DDR3
Rede	1 Gigabit
Disco	1TB Sata III 3Gbps
Sistema Operacional	Ubuntu 12.04 64-bit
Transcoder	FFmpeg 1.0
Bibliotecas	libx264 e libfdk_aac
Armazenamento	Partição NFS no Mestre

Cada nó do cluster possui dois processadores Intel com 4 núcleos, habilitados para a execução de duas threads por núcleo, oferecendo um total de 8 processadores físicos e a execução de 16 threads simultâneas. As Tabelas 1 e 2 mostram os parâmetros do ambiente experimental.

O vídeo utilizado [2] estava codificado em H.264 com a taxa de 1200 kbps e resolução de 1280 x 720 pixels. Para o áudio foi usado a codificação AAC , mantendo a taxa original de 128 kbps. A transcodificação converteu o vídeo para uma taxa de 800kbps e manteve a mesma resolução. Nas duas instâncias o áudio e o vídeo foram sincronizados no conteiner MP4 [4].

Tabela 2: Ambiente Experimental - Vídeo

Formato	MP4
Codecs	H.264 e AAC
Resolução	1280x720 px
Bitrate de Vídeo	1200 kbps
Bitrate de Áudio	128 kbps
Duração	4min 25s

Nos experimentos utilizamos o software FFmpeg [1] com libx264 para implementar a análise dos metadados, transcodificação e a integração dos segmentos transcodificados. O FFmpeg permite através da opção "seek"fazer a busca de um quadro I em um ponto desejado de um arquivo de vídeo.

Em todos os experimentos realizados, utilizamos os resultados obtidos da média de 30 execuções. Avaliamos o tempo de transcodificação em um nó do cluster. O desempenho da transcodificação, no modo *multithreading*, é mostrado no gráfico da Figura 3. Em uma máquina de 8 cores, observamos um desempenho 5 vezes melhor com 48 threads. As quantidades maiores de threads apresentaram resultados próximos, mas com variação muito alta do tempo de execução.

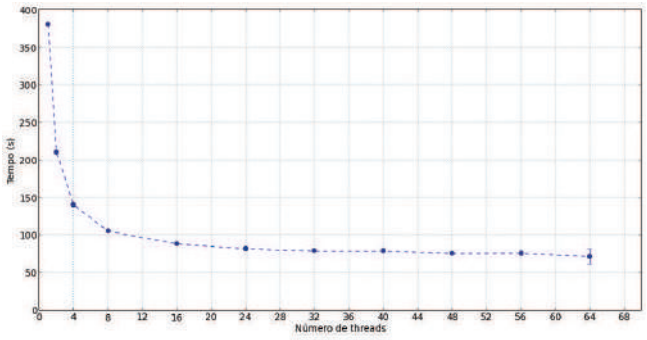

Figura 3: Tempo da Transcodificação em 1 Nó com 8 Cores.

O gráfico de barras na Figura 4 apresenta o resultado da execução do sistema de transcodificação distribuída no ambiente experimental. Um processo FFmpeg, executando 16 threads, foi escalonado em cada nó para a transcodificação de um segmento de 1/4 do vídeo. No referido gráfico encontramos o tempo de transcodificação de cada segmento e o tempo total, composto pelos tempos decorridos na análise/segmentação, transcodificação e a integração do áudio e segmentos de vídeo no formato MP4. A Tabela 3 apresenta o tempo decorrido em cada componente da barra Total. O tempo de transcodificação do áudio, decorrido concorrentemente à transcodificação do vídeo, foi de 9,03 segundos.

O resultado apresentado demonstra a eficiência do sistema proposto. O tempo de conversão do vídeo foi 27,92 segundos, que representa um ganho de desempenho de 2,7 vezes se comparado com melhor resultado do modo *multithreading*. Podemos observar no gráfico a ocorrência de desbalanceamento de carga no escalonamento dos processos de transcodificação. A razão é a variação da complexidade na compressão de diferentes

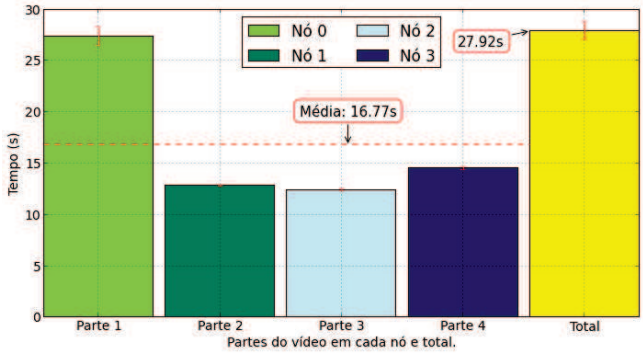

Figura 4: Tempo da Transcodificação Distribuída - 4 Segmentos em 4 Nós com 16 Threads por Segmento

Tabela 3: Componentes do Tempo Total de Transcodificação.

Análise	Transcodificação	Integração	Total
0.19s	27.36s	0.37s	27.92s

trechos do vídeo. Devido ao desbalanceamento de carga, a média do tempo de transcodificação de cada segmento

foi 16,77 segundos. Este resultado sugere que podemos aumentar o desempenho do sistema se melhoramos o escalonamento das tarefas.

4.1 Threads e Processos

Tendo em vista o desbalanceamento de carga ocorrido no experimento anterior, procuramos diminuir a ociosidade dos nós aumentando o número de segmentos de vídeo para transcodificação. Neste experimento exploramos o uso dos recursos do nó de transcodificação variando a quantidade de threads e processos. Executamos de 8 a 64 threads por nó e foram escalonados de 1 a 8 processos de transcodificação simultaneamente.

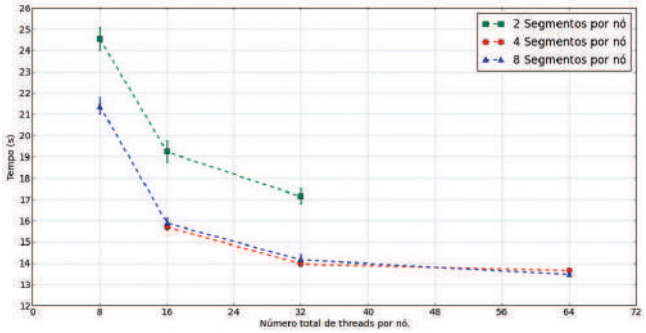

Figura 5: Tempo da Transcodificação Distribuída vs. No. Segmentos - Cluster com 4 Nós

Os resultados são apresentados no gráfico da Figura 5. O menor tempo obtido - 13,5 segundos-, foi na transcodificação usando 8 segmentos por nó e 8 threads por processo. Observamos no gráfico que o aumento da quantidade de segmentos contribui mais para melhorar o desempenho da transcodificação distribuída do que o aumento do número de threads em cada nó. A razão deve-se ao aumento do número de segmentos e em consequência a dispersão dos trechos de codificação mais complexos entre os nós, diminuindo o desbalanceamento da carga, como podemos observar no gráfico da Figura 6. O gráfico mostra os tempos de transcodificação de cada segmento para a melhor composição de segmentos e threads. Observamos a melhora no desbalanceamento da carga entre os nós, o que consequentemente contribuiu para o aumento do desempenho do sistema de transcodificação distribuído. Em comparação com o melhor caso no modo multithreading, onde havia 48 threads, o processamento é feito 5,6 vezes mais rápido com uso de apenas 4 vezes mais recursos de hardware.

Tabela 4: Componentes do tempo total de transcodificação da Figura 6.

Análise	Transcodificação	Integração	Total
0.19s	12.93s	0.37s	13.49s

5. CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho descreveu e analisou um sistema de transcodificação distribuída, que propõe uma técnica genérica de segmentação de vídeos codificados com compressão temporal. Esta característica é encontrada nos principais codecs do mercado, como H.264/AVC e VP8 por exemplo. Os resultados encontrados demonstram que a solução consegue

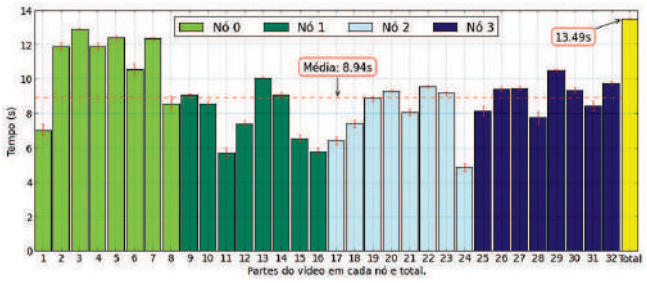

Figura 6: Tempo da Transcodificação Distribuída - 32 Segmentos em 4 Nós com 8 threads por segmento

acelerar a transcodificação de vídeo em 5,6 vezes em um cluster de 4 nós dual quad-core quando comparado com a melhor execução (48 threads) da mesma transcodificação em um nó apenas.

Uma barreira encontrada para o desempenho do sistema proposto foi o desbalanceamento de carga na transcodificação de cada segmento. Mostramos que o uso de um número de segmentos maior que a quantidade de nós no cluster permite distribuir melhor a carga no sistema. Devemos realizar outras avaliações para esta solução, como explorar a granularidade dos segmentos, variar a características dos vídeos e aumentar o número de nós no cluster em trabalhos futuros.

Reconhecimento

Este trabalho foi desenvolvido dentro do projeto RSVP, financiado pelo Convênio FINEP-COPPETEC 01.09.0570.00. João Bernado Vianna S. de M. de Oliveira é aluno de Iniciação Científica e principal autor. Claudio L. de Amorim, Lauro L. A. Whately e Diego L. C. Dutra colaboraram e orientaram este trabalho.

6. REFERÊNCIAS

[1] FFmpeg. Ffmpeg 1.0. `http://ffmpeg.org/`, 2013. Acessado em Março de 2013.

[2] B. Hammack. Ccd: The heart of a digital camera. `http://www.youtube.com/watch?v=wsdmt0De8Hw`, 2012. Acessado em Março de 2013.

[3] Z. Huang, C. Mei, L. Li, and T. Woo. Cloudstream: Delivering high-quality streaming videos through a cloud-based svc proxy. In *INFOCOM, 2011 Proceedings IEEE*, pages 201–205, 2011.

[4] ISO. *ISO/IEC 14496-12:2012: Information technology – Coding of audio-visual objects – Part 12: ISO base media file format*. 2012. Available in English only.

[5] R. Pereira and K. Breitman. An architecture for distributed high performance video processing in the cloud. 2010.

[6] I. E. G. Richardson. *H.264 and MPEG-4 Video Compression: Video Coding for Next Generation Multimedia*. John Wiley & Sons, Ltd, 2004.

[7] Y. Sambe, S. Watanabe, D. Yu, T. Nakamura, and N. Wakamiya. High-speed distributed video transcoding for multiple rates and formats. *IEICE - Transactions on Information and Systems archive*, Volume E88-D Issue 8, 2005.