

Sistema de Distribuição de GOPs

Eduardo Flohr,¹ Rafael R. Obelheiro,¹ Maurício A. Pillon¹

¹Departamento de Ciência da Computação – DCC
Universidade do Estado de Santa Catarina – UDESC

eduardo.flohr@gmail.com, {rro, mpillon}@joinville.udesc.br

1. Introdução

Com a facilidade de se adquirir filmadoras, gravadores e software de edição de vídeo, a quantidade de dados armazenados na forma de vídeo cresceu muito ultimamente, mas o poder de processamento de computadores pessoais não é suficiente para o processamento de vídeos, que pode ser definido também como a aplicação de técnicas de tratamento de vídeo [Tahaghoghi et al. 2005].

Pesquisadores que desenvolvem novas técnicas de tratamento de imagem e vídeo, como a detecção de objetos ou aplicação de efeitos, também encontram dificuldades para testá-las, pois o poder de processamento necessário para isso é alto [Barbosa et al. 2000]. Um vídeo pode ser interpretado como uma coleção de imagens, que são chamados de *frames*, que possuem uma relação temporal e espacial, podendo ser fracionado nessas duas dimensões [Guo et al. 2007]. Na dimensão temporal, utilizada neste trabalho, o fracionamento é feito através da divisão do vídeo em períodos de tempo, criando grupos de *frames* sequenciais [Missaoui e Palenichka 2004]. Um período de tempo é chamado de grupo de imagens (GOP, *Group of Pictures*). O fracionamento do vídeo permite a aplicação de técnicas de processamento de imagens em cada *frame*, tendo como resultado um vídeo manipulado por um algoritmo de tratamento de imagem. Tanto o tratamento de vídeo quanto o tratamento de imagens possuem limitações em relação ao poder de processamento, podendo ser aprimorados utilizando um sistema distribuído, já que através dele é possível reduzir o tempo de execução de tarefas computacionais [Watanabe et al. 2003].

Aplicado ao processamento de vídeo, um sistema distribuído pode propor dois elementos: a possibilidade de incrementar o tamanho do vídeo que será processado, e o aumento no poder computacional para realizar o processamento. Uma plataforma distribuída que possibilita esses elementos é o *cluster*. Nessa plataforma, nós se comunicam entre si através de trocas de mensagens, que podem ser controladas pelo próprio desenvolvedor, ou por bibliotecas específicas, tais como MPI (*Message Passing Interface*) [MPI Forum 1994] e PVM (*Parallel Virtual Machine*) [Geist et al. 1994]. Dentre as bibliotecas estudadas, escolheu-se a MPI devido ao suporte à linguagem C e à familiaridade dos desenvolvedores.

A utilização de bibliotecas de troca de mensagens permite que o programador se abstraia de algumas tarefas como o controle da camada de transporte e a interoperabilidade do sistema [MPI Forum 1994], se preocupando apenas com o conteúdo das mensagens e com o escalonamento do sistema. O escalonamento pode ser definido como o gerenciamento de recursos com foco em distribuir de forma transparente e clara a carga do sistema entre seus componentes. Nesse contexto, este trabalho apresenta uma arquitetura distribuída para divisão, processamento distribuído de GOPs e concatenação de vídeo em

um *cluster*. Algoritmos de processamento de imagens sequenciais são aplicados a porções de vídeos distribuídos em nós do *cluster*.

Este artigo está dividido em: Introdução, cujo objetivo é contextualizar o problema tratado; Sistema de Distribuição de GOPs, onde estão descritos as características e o esquema de distribuição; Resultados Parciais, onde encontra-se a descrição do ambiente de teste e os resultados parciais; e, finalmente, a Conclusão.

2. Sistema de Distribuição de GOPs

A distribuição de algoritmos de processamento de imagens e, conseqüentemente, de processamento de vídeo não é uma tarefa trivial. Isso ocorre, principalmente, pelo fato do desenvolvedor ser obrigado a possuir conhecimento em duas áreas distintas, processamento de imagens e sistemas distribuídos. O Sistema de Distribuição de GOPs tem por objetivo aplicar algoritmos sequenciais de processamento de imagens em porções de vídeos, responsabilizando-se pela parte de comunicação, distribuição de GOPs e manipulação de vídeos. O modelo (figura 1) escolhido possui três componentes: Servidor Coordenador, Servidor Auxiliar, e o Nó.

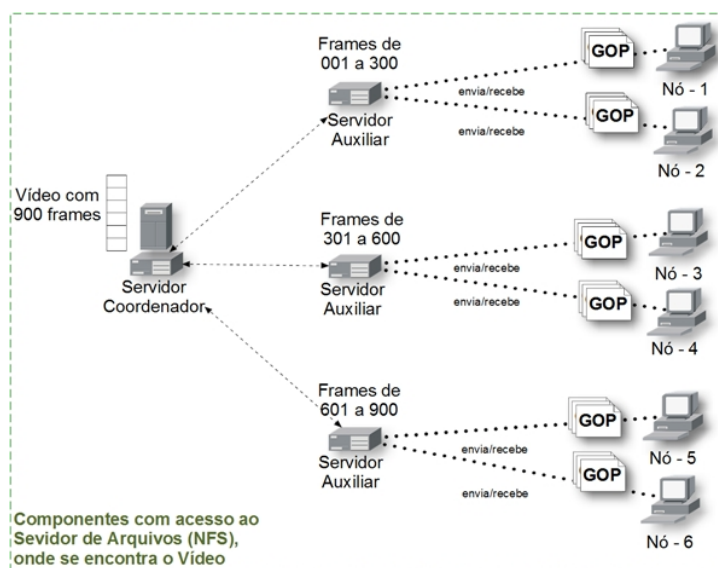


Figura 1. Arquitetura do sistema.

O **Servidor Coordenador** é responsável pela interface com o usuário, o escalonamento do vídeo entre os Servidores Auxiliares e a união dos sub-vídeos em um único vídeo novamente. Todos os processos no *cluster* possuem acesso a um Servidor de Arquivos onde se encontra o vídeo a ser processado. O **Servidor Auxiliar** é responsável por fracionar a porção do vídeo, que lhe foi atribuído pelo Servidor Coordenador, em GOPs, escalonar os GOPs para os Nós e unir os GOPs em forma de vídeo novamente. O **Nó** é responsável por processar os GOPs que receber do Servidor Auxiliar. Esse processamento é realizado aplicando-se uma técnica de tratamento de imagem sobre cada *frame* dos GOPs que o Nó receberá, retornando o resultado dos GOPs processados ao Servidor Auxiliar. Após o envio do resultado dos GOPs processados o Nó ficará disponível para receber outros GOPs.

O sistema proposto é simples, mas apresenta alguns desafios: (1) granularidade das tarefas; (2) escalonamento; (3) balanceamento de carga. Inicialmente, desconsiderou-se a problemática vinculada à granularidade das tarefas. O algoritmo de escalonamento aplicado foi o de dividir o número de *frames* pelo número de Servidores Auxiliares. Finalmente, o balanceamento de carga se dá pela distribuição de tarefas aos nós sob demanda: os servidores auxiliares possuem um conjunto de tarefas, que são enviadas aos seus nós quando estes estão ociosos (ao encerrar a tarefa anterior, por exemplo). Deve-se observar que não há política de balanceamento de carga entre as tarefas dos servidores auxiliares.

3. Resultados Parciais

Para a realização dos testes foi utilizado um ambiente com 15 computadores homogêneos interligados através de uma rede 100 Mbps. O servidor de arquivos, presente em outro computador com as mesmas configurações, encontrava-se na mesma rede local. Cada computador é formado por: quatro processadores AMD Phenom II X4 B93 2800 MHz, cache L1 de 512 KB (128 KB por core), cache L2 de 2 MB (512 KB por core), memória RAM de 4 GB, placa de rede Broadcom NetXtreme BCM5761 Gigabit Ethernet PCIe e disco SATA/UDMA/133 de 500 GB com 7.200 rpm. Os softwares utilizados foram: Linux Debian com kernel 2.6.26-2-686, GCC 4.3.2, Open MPI 1.2.7rc2, FFMPEG r11872 e OPENCV 2.1. Em todos os testes foram utilizados um vídeo de 50 minutos de duração, que possui um total de 75.000 *frames*. Ao longo do vídeo original, incluiu-se 1.800 *frames* com círculos. Esses círculos foram introduzidos no vídeo com o intuito de testar a corretude do vídeo de saída no caso de uso da técnica de segmentação de imagens. Assim, o resultado esperado da aplicação da técnica escolhida (detecção de formas) no vídeo utilizado nos testes é um vídeo de apenas 3 segundos contendo somente os *frames* com círculos.

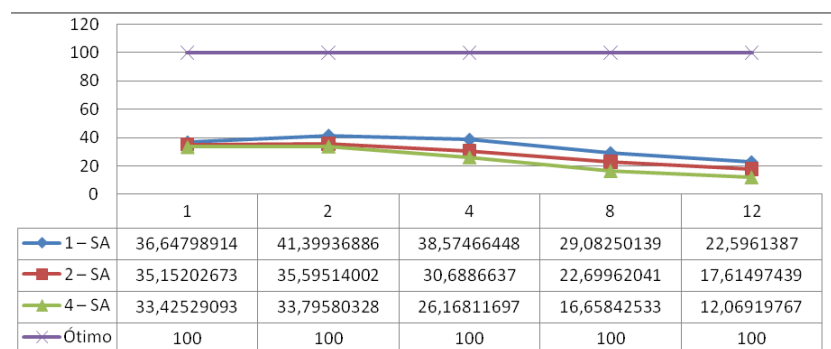


Figura 2. Eficiência do sistema em relação a quantidade de Nós por Servidor Auxiliar.

A figura 2 apresenta o gráfico da eficiência relacionada à quantidade de Nós utilizados por cada Servidor Auxiliar. O melhor resultado obteve uma eficiência de 41,40% com somente um Servidor Auxiliar e 2 Nós de processamento. Pode-se ainda observar que, em cada uma das três distribuições utilizadas de Servidores Auxiliares, o melhor valor de eficiência encontrado foi com a utilização de dois Nós para cada Servidor Auxiliar. É visível na figura um aumento da eficiência em relação à utilização de apenas um Nó para cada Servidor Auxiliar, e uma diminuição da eficiência na utilização de quatro, oito e doze Nós para cada Servidor Auxiliar em relação à utilização de dois Nós. Esses baixos

valores de eficiência obtidos com a técnica de tratamento utilizada podem ser justificados pelo fato do esforço necessário para a aplicação da técnica, no caso a detecção de formas, ser pequeno em relação ao poder computacional da plataforma e ao esforço de envio dos *frames*.

4. Conclusão

Técnicas de processamento de imagens distribuídas exigem do desenvolvedor conhecimentos associados a duas áreas distintas, tornando-as complexas. Neste trabalho foi modelado e implementado um sistema de distribuição de GOPs com foco na divisão, distribuição e manipulação de vídeos, sem a necessidade de conhecimentos específicos dos algoritmos de processamento de imagens. O desenvolvimento do sistema foi realizado através da utilização da biblioteca MPI para realizar a comunicação entre os processos do sistema. A principal indicação deste sistema é a sua utilização onde seja necessária a aplicação, sobre um vídeo, de técnicas de tratamento de imagens que necessitem de um grande poder computacional para o seu processamento. Uma das áreas que se encaixa nesse requisito é a área de segurança, onde se torna necessária a utilização de técnicas complexas de identificação de pessoas e placas em vídeos de longa duração. Através dos testes realizados no sistema, foi possível comprovar a redução de tempo no processamento de um vídeo comparado à sua execução na forma sequencial, mas a eficiência dos testes realizados não foi satisfatória porque as técnicas de tratamento utilizadas não necessitam de um alto poder computacional para serem aplicadas aos *frames*, não consumindo todo o poder computacional disponível na plataforma *cluster* utilizada.

Referências

- Barbosa, J. M. G., Tavares, J. M. R. S., e Padilha, A. (2000). Parallel image processing system on a cluster of personal computers. *4th International Conference on Vector and Parallel Processing*.
- MPI Forum. (1994). A message-passing interface standard.
- Geist, A., Beguelin, A., Dongarra, J., Jiang, W., Manchek, R., e Sunderam, V. (1994). *PVM: Parallel Virtual Machine*. The MIT Press, 1st Edition.
- Guo, M., Lu, Y., Wu, F., Li, S., e Gao, W. (2007). Distributed video coding with spatial correlation exploited only at the decoder. *IEEE International Symposium on Circuits and Systems*, pp. 41–44.
- Missaoui, R. e Palenichka, R. M. (2004). Effective image and video mining: an overview of model-based approaches. *Sixth International Workshop on Multimedia Data Mining: Mining Integrated Media and Complex Data*.
- Tahaghoghi, S. M. M., Williams, H. E., Thom, J. A., e Volkmer, T. (2005). Video cut detection using frame windows. *Proceedings of the Twenty-eighth Australasian conference on Computer Science – Volume 38*.
- Watanabe, H., Ghatpande, A., e Nakazaka, H. (2003). Distributed computing for real-time video processing. *First International Conference on Ubiquitous Computing*.