

Um Protótipo de Servidor Multimídia com Mecanismos de QoS *

Adriane Q. Cardozo¹, Edmundo de Souza e Silva¹, Rosa M. M. Leão¹

¹Universidade Federal do Rio de Janeiro - COPPE/Prog. de Engenharia de Sistemas e Computação
Caixa Postal 68511, CEP: 21941-972–Rio de Janeiro, RJ
drika, edmundo, rosa@land.ufrj.br

Aplicações em Redes Avançadas: Ensino a Distância, Vídeo Digital

1. Introdução e Contribuições

As aplicações de mídia contínua consistem de uma seqüência de dados (geralmente, amostras de áudio ou quadros de vídeo) que devem ser apresentados em um determinado intervalo de tempo. Devido à natureza contínua de seus dados e à alta interatividade, essas aplicações impõem limites na busca e na entrega de dados, pois perdas e atrasos podem causar interrupções que comprometem a qualidade da apresentação.

O objetivo deste trabalho foi o estudo e o desenvolvimento de uma classe de sistemas multimídia. Para isto foram abordados os problemas da rede, devido ao tipo de serviço oferecido pela Internet (*best effort*), e algumas soluções encontradas; e os aspectos do projeto de servidores multimídia, entre eles, políticas de acesso aos dados, organização dos dados, escalonamento dos discos, controle de admissão de usuários e técnicas para compartilhamento de recursos [Gemmell et al., 1995, Vin et al., 1994, de Souza e Silva et al., 2002]. Nosso trabalho é baseado no servidor RIO [Santos, 1998] desenvolvido na UCLA. Foi feita a proposta e a implementação de um conjunto de mecanismos incluindo: (i) gerenciamento de *buffers* no servidor e no cliente, para compensar as flutuações da carga dos discos e da rede, visando o atendimento de um maior número de usuários, e (ii) um novo controle de admissão estatístico, visando a garantia da qualidade de serviço para os clientes. Como parte do trabalho, também foi criado um modelo do servidor RIO com o uso da ferramenta Tangram-II [LAND, 2001]. Através de simulações, várias medidas puderam ser obtidas que auxiliaram na configuração do protótipo e entendimento do sistema. O servidor desenvolvido neste trabalho será utilizado no curso de Tecnólogo em Informática do consórcio CEDERJ [CEDERJ, 2002] para armazenamento e distribuição dos vídeos das aulas. Para isto foi desenvolvido pelo grupo de pesquisa LAND um novo cliente de vídeo com suporte para sincronização de *slides*.

2. Gerenciamento de Buffers e Controle de Admissão de Usuários

O servidor RIO (*Randomized I/O Multimedia Storage Server*) é um sistema de armazenamento paralelo multimídia universal, baseado em alocação aleatória e replicação de blocos. Ele suporta diversos tipos de objetos e é capaz de gerenciar múltiplas aplicações concorrentes, com ou sem restrições de tempo. Os principais motivos para a utilização do RIO foram a flexibilidade da alocação aleatória para armazenamento e acesso aos objetos, a exploração de uma nova e promissora arquitetura e os projetos em conjunto com a UCLA e a UFMG, além do servidor

*Este trabalho é parcialmente financiado pelo CNPq e FAPERJ.

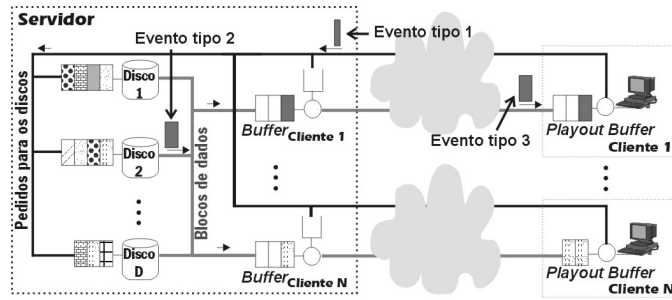


Figura 1: Visão do Sistema

suportar o armazenamento de diversos tipos de objetos, essencial para a aplicação alvo: ensino a distância, em particular os projetos CEDERJ e MAPPED.

O tipo de aplicação abordado é a transmissão de áudio e vídeo sob demanda. Supondo que o padrão de acesso típico destas aplicações é seqüencial, é possível utilizar as informações *a priori* dos vídeos armazenados para prever o escalonamento dos pedidos de cada cliente durante a sua execução. Na literatura já existem algoritmos que exploram esta idéia. Uma das diferenças do controle de admissão proposto é que este não considera o atendimento dos pedidos baseado em ciclos, pois o servidor RIO não faz o atendimento em ciclos. Além das informações *a priori* dos vídeos requisitados, o controle de admissão é baseado em medidas de desempenho do servidor calculadas em tempo real, como por exemplo, tempo de serviço dos discos, e nas medidas da rede (RTT). Desta forma foi possível criar uma infra-estrutura, com um controle detalhado do sistema, para estudo e coleta de medidas, tais como a capacidade do servidor, o tempo de serviço dos discos, o tamanho do *playout buffer* dos clientes e o RTT entre o servidor e cada um dos clientes. Com isto, vários experimentos podem ser realizados de forma a ajustar tanto os parâmetros do servidor, quanto os parâmetros dos clientes para alcançar os níveis de QoS desejados.

Para compensar as flutuações na carga dos discos, foi proposto um gerenciamento de *buffers* no servidor. Existe um *buffer* no servidor para cada cliente (Figura 1), onde são armazenados blocos, não apenas para o envio imediato ao cliente (blocos já requisitados), mas também para o armazenamento dos blocos cuja leitura é feita antes de sua solicitação por parte do cliente (*prefetching*), utilizando a técnica *Fixed Lookahead*. Juntamente com o gerenciamento de *buffers*, o controle de admissão proposto faz uma análise em tempo real para decidir se um novo cliente pode ser admitido ou não. Esta análise é baseada na previsão dos pedidos de blocos dos clientes já admitidos e nas informações de desempenho do servidor (extraídas em tempo real). A cada novo pedido de admissão, é feita a simulação de todo o sistema (Figura 1), incluindo o novo cliente, desde a chegada das requisições no servidor (evento tipo 1) até o recebimento dos dados em cada cliente (evento tipo 3). O evento tipo 2 simula a leitura de um bloco de dados no disco. O novo cliente é admitido se a QoS puder ser garantida para todos os clientes. Os parâmetros analisados são o número máximo de *hiccups* devido ao esvaziamento do *playout buffer* do cliente e o intervalo máximo que este *buffer* pode ficar vazio.

O modelo criado, Figura 2, é um modelo simplificado que representa o funcionamento do servidor e o gerenciamento de *buffers*. A distribuição utilizada para modelar o tempo de serviço dos discos foi a distribuição Normal, com os parâmetros média e variância do tempo de serviço para a leitura de um bloco de dados. Em algumas simulações também foi utilizada a distribuição Lognormal para verificar a sensibilidade do modelo. A partir de simulações foram extraídas medidas como o tamanho médio das filas dos discos de acordo com a carga gerada

pelos clientes, a quantidade de clientes que podem ser admitidos considerando o comportamento do objeto Novo Cliente e a quantidade de falhas no novo cliente para um dado tamanho de *playout buffer*. Estas medidas auxiliam, por exemplo, na configuração do tamanho do *playout buffer* no cliente e na configuração do tamanho do *buffer* para cada cliente no servidor, de acordo com a QoS desejada. Com isto, pode-se observar o ganho obtido com a utilização de *buffers* no servidor para cada cliente, o impacto da carga gerada por todos os usuários, de acordo com a limitação das taxas dos discos, e do retardo da rede, entre o servidor e o novo candidato à admissão.

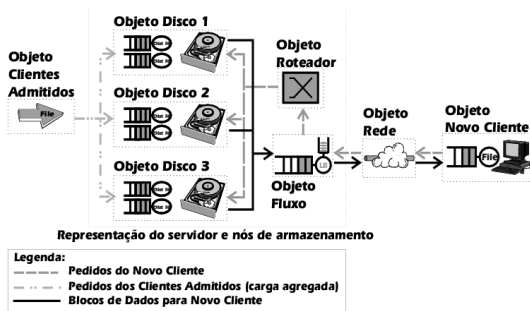


Figura 2: Modelo do servidor

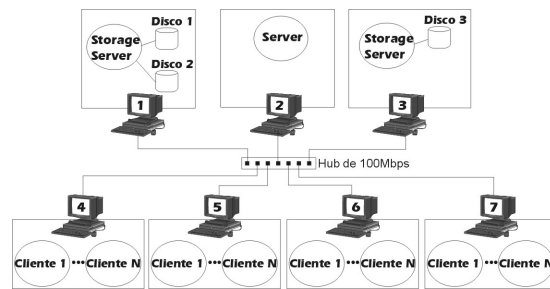


Figura 3: Plataforma dos testes

Para a avaliação dos mecanismos implementados no servidor multimídia foram realizados vários experimentos, com isto, diversas medidas foram obtidas do protótipo, tais como o tempo de espera e o tempo de serviço dos discos utilizados, o tempo de geração da lista de requisições para cada cliente, o tempo de geração da lista de eventos para o processo de admissão, o tempo de execução do controle de admissão e o RTT entre o servidor e os clientes. A plataforma usada está ilustrada na Figura 3. Para a geração da carga de trabalho foi implementado um programa em linguagem C que dispara vários clientes.

Os resultados obtidos a partir das simulações do modelo foram comparados com os resultados dos experimentos e com o uso de um controle de admissão simples, conforme ilustra a Tabela 1. No controle de admissão simples foram calculadas três tipos de média para o tempo de serviço dos discos. O tipo 1, tempo médio, é a média aritmética e o tipo 2, tempo estimado, é a média que se move exponencialmente (EWMA) das amostras coletadas nos experimentos para o tempo de serviço dos discos. Já o tipo 3 utiliza os tempos coletados com a ferramenta *hdparm*. Cabe salientar que a diferença dos experimentos 1 ao 6 para os experimentos 7 ao 12 foi o aumento no tamanho do *buffer* de cada cliente, e destes experimentos para o experimento 13 foi que neste último o controle de admissão foi desligado (servidor atendeu todos os clientes disparados). A descrição completa dos mecanismos e dos resultados dos experimentos e das simulações pode ser encontrada na dissertação [de Quevedo Cardozo, 2002].

3. Conclusões e Trabalhos Futuros

Neste trabalho foi proposto e implementado um conjunto de mecanismos que visam a melhoria do desempenho e utilização do servidor multimídia, além da garantia de QoS para os usuários do sistema. Tais mecanismos foram incorporados no servidor RIO. Vários experimentos foram realizados de modo a verificar o desempenho do servidor com os novos recursos implementados. Em paralelo, foi desenvolvido um modelo simplificado do servidor e diversas medidas foram obtidas através de simulações. Foi observado através dos resultados dos experimentos que o controle implementado permitiu garantir a QoS dos clientes e também que o aumento

		Número de Clientes	
		Admitidos	QoS não garantida
Controle de admissão implementado	Experimento 1 ao 6	68 - 73	0 - 3
(controle fino)	Experimento 7 ao 12	75 - 85	0
	Experimento 13	90	90
Controle de admissão simples	tempos médios - tipo 1	74	sem garantia
(utilizando tempos coletados)	tempos estimados - tipo 2	99	sem garantia
	tempos coletados com <i>hdparm</i> - tipo 3	126	sem garantia
Simulações do modelo	Dist. Normal	111	0
(utilizando tempos estimados)	Dist. Lognormal	101	0

Tabela 1: Resumo dos resultados

dos buffers tanto no servidor quanto no cliente permitiram a admissão de um número maior de clientes. Como resultado final, foram feitas diversas melhorias no RIO, implementando novas funcionalidades e um ambiente para estudo e coleta de dados. É importante enfatizar que este trabalho não foi apenas experimental, mas também serviu de base para um grande empreendimento no Estado do Rio de Janeiro: ensino a distância - projeto CEDERJ. Como trabalhos futuros pode-se citar, por exemplo, o compartilhamento dos *buffers* no servidor para a redução de pedidos aos discos e o atendimento de um maior número de usuários; o estudo de outros aspectos no modelo; o uso de mecanismos para alterar a qualidade do vídeo de acordo com os recursos disponíveis e o uso de *proxy* e de técnicas de compartilhamento de recursos para redução do tráfego na rede. Atualmente, já está em fase de implementação a técnica de *patching* no servidor RIO.

Referências

- CEDERJ (2002). Centro de Educação de Ensino Superior a Distância do Estado do Rio de Janeiro. URL <http://www.cederj.rj.gov.br>.
- de Quevedo Cardozo, A. (2002). Mecanismos para Garantir Qualidade de Serviço de Aplicações de Vídeo sob Demanda. Master's thesis, COPPE - UFRJ (*Universidade Federal do Rio de Janeiro*).
- de Souza e Silva, E., Leão, R. M. M., Ribeiro-Neto, B., and Campos, S. (2002). Performance Issues of Multimedia Applications. In *Performance Evaluation of Complex Systems: Techniques and Tools*.
- Gemmell, D. J., Vin, H., Kandlur, D. D., and Rangan, P. V. (1995). Multimedia Storage Servers: A Tutorial. In *IEEE Computer*, pages 40–49.
- LAND (2001). *Laboratory for modeling, analysis and development of networks and computing systems*. URL <http://www.land.ufrj.br>.
- Santos, J. R. G. (1998). *RIO: A Universal Multimedia Storage System Based on Random Data Allocation and Block Replication*. PhD thesis, UCLA.
- Vin, H. M., Goyal, A., Goyal, A., and Goyal, P. (1994). An Observation-Based Admission Control Algorithm for Multimedia Servers. In *Proceedings of the First IEEE International Conference on Multimedia Computing and Systems (ICMCS)*, pages 234–243.