

UM ESTUDO DO SERVIDOR RIO EM UMA REDE DE ALTA VELOCIDADE

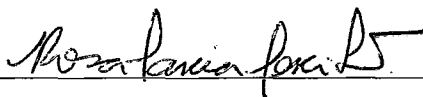
Marcello Ribeiro Valle

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

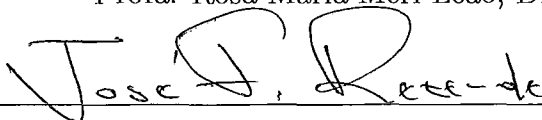
Aprovada por:



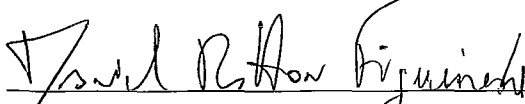
Prof. Edmundo Albuquerque de Souza e Silva, Ph.D.



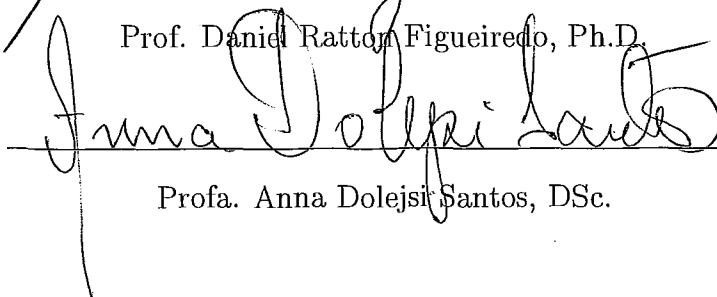
Profa. Rosa Maria Meri Leão, Dr.



Prof. José Ferreira de Rezende, Dr.



Prof. Daniel Rattón Figueiredo, Ph.D.



Profa. Anna Dolejsi Santos, DSc.

RIO DE JANEIRO, RJ - BRASIL

AGOSTO DE 2007

VALLE, MARCELLO RIBEIRO

Um estudo do servidor RIO em uma rede de alta velocidade [Rio de Janeiro] 2007

XVII, 95 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2007)

Dissertação - Universidade Federal do Rio de Janeiro, COPPE

1. Educação a Distância
2. Avaliação de Desempenho
3. Qualidade de Serviço
4. Servidores Multimídia
5. Redes de Computadores

I. COPPE/UFRJ II. Título (Série)

Agradecimentos

Ao fim de mais uma etapa não posso deixar de agradecer a todos do LAND - Laboratório de Modelagem, Análise e Desenvolvimento de Redes e Sistemas de Computação. Em especial aos professores Rosa Maria Meri Leão e Edmundo A. de Souza e Silva que me orientaram durante este trabalho.

Agradeço aos meus pais, Orlando e Marlene, pelo apoio e encorajamento, que foram fundamentais para que eu chegasse até aqui. Também não posso esquecer da minha namorada, Fabiana, pela paciência e compreensão durante os momentos de ausência.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

UM ESTUDO DO SERVIDOR RIO EM UMA REDE DE ALTA VELOCIDADE

Marcello Ribeiro Valle

Agosto/2007

Orientadores: Edmundo de Souza e Silva

Rosa Maria Meri Leão

Programa: Engenharia de Sistemas e Computação

O trabalho apresentado nessa dissertação é um estudo sobre o servidor multimídia RIO em uma rede de alta velocidade usando uma carga real de usuários interativos. Foram coletados dados de experimentos reais relativos à qualidade de serviço percebida pelos clientes bem como ao tráfego gerado por um dos nós de armazenamento. Com essas medidas foi possível avaliar o impacto de escolhas de projeto tais como: a utilização de *cache* pelos clientes, a replicação de conteúdo do servidor e a distribuição de nós de armazenamento em redes heterogêneas. Foi realizada também uma comparação entre os resultados obtidos com os experimentos com uma carga de trabalho real e uma sintética, gerada por um modelo desenvolvido em um trabalho paralelo a este. As principais conclusões foram: A adição de um *storage server* em uma rede inferior as demais pode trazer ganhos para o sistema em alguns cenários; apesar do bom balanceamento de carga promovido pela técnica de alocação aleatória de blocos é possível que ocorra o desbalanceamento de curto prazo, o que pode aumentar a latência do sistema, porém, a replicação do conteúdo pode minimizar o desbalanceamento de curto prazo; os resultados obtidos com a carga sintética foram bem próximos dos obtidos com carga real, o que demonstra a boa acurácia do modelo.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

A STUDY OF RIO SERVER ON A HIGH SPEED NETWORK

Marcello Ribeiro Valle

August/2007

Advisors: Edmundo de Souza e Silva
Rosa Maria Meri Leão

Department: Computer and System Engineering

The work performed in this dissertation is the performance study of the multimedia server RIO on a high speed network submitted to a real interactive workload. We collected data necessary to evaluate the QoS perceived by the users as well as the traffic originated from the storage nodes. From the collected measures we evaluated the impact of different server configurations such as: a) the use of cache in the client nodes; b) content replication and c) distribution of storage nodes through heterogeneous networks. We have also compared the results obtained from experiments using a real workload and a synthetic workload, generated by a model developed in another work. The most relevant conclusions are: a) the addition of a storage server located in the regular Internet (not the giga net) can improve the performance in some scenarios, b) despite the load balancing resulting from the random data allocation, we can observe short-term unbalancing that may increase the system's latency. However, content replication can minimize short-term unbalancing. The results obtained from the use of synthetic workload are almost identical to those obtained from the real workload and therefore we conclude that the synthetic workload can be used in the large scale stress performance experiments of the RIO server.

Sumário

Resumo	iv
Abstract	v
Glossário	xvi
1 Introdução	1
1.1 Objetivo e Motivação	2
1.2 Contribuição	4
1.3 Organização do trabalho	5
2 Aplicações Multimídia na Internet	6
2.1 Classificações quanto a natureza do serviço prestado	7
2.1.1 Transmissão de Áudio e Vídeo Pré-Armazenados	7
2.1.2 Transmissão de Áudio e Vídeo ao Vivo	8
2.1.3 Transmissão de Áudio e Vídeo Interativos em Tempo Real	8
2.2 Classificações quanto a distribuição dos componentes	8
2.2.1 VoD	8
2.2.2 DVoD	9

Proxies	10
Redes de distribuição de conteúdo	12
2.2.3 VoD Cooperativo	13
2.3 Classificações quanto ao nível de interatividade oferecida	14
2.4 Arquitetura Geral de um servidor multimídia	15
2.4.1 Armazenamento em Disco	16
2.4.2 Tolerância a Falhas de Disco	19
2.4.3 Múltiplos Caminhos	20
2.4.4 Considerações do lado Cliente: <i>Buffers</i> , Latência Inicial e <i>Cache</i>	22
3 Servidor Multimídia RIO	23
3.1 Visão Geral	23
3.2 Visão Detalhada do servidor RIO	25
3.2.1 Gerenciador	27
Gerenciador de Sessão (<i>Session Manager</i>)	27
Gerenciador de Fluxos (<i>Stream Manager</i>)	27
Gerenciador de Eventos (<i>Event Manager</i>)	27
Gerenciador de Objetos (<i>Object Manager</i>)	28
Gerenciador de Disco (<i>Disk Manager</i>)	28
Roteador (<i>Router</i>)	28
3.2.2 Servidor de armazenamento	29
Interface com o Roteador (<i>RouterInterface</i>)	29
Dispositivo de Armazenamento (<i>StorageDevice</i>)	30

Gerenciador de Armazenamento (<i>StorageManager</i>)	30
Interface com o Cliente (<i>ClientInterface</i>)	31
3.2.3 Controle de admissão	31
3.2.4 Policiamento de pedidos	32
3.3 Clientes RIO	32
3.3.1 Cliente Riosh	32
3.3.2 Cliente <i>RioMMClient</i>	33
3.3.3 Emulador de clientes	35
4 Ambiente de Testes	38
4.1 Rege Giga	38
4.2 Projeto DIVERGE	39
4.2.1 Equipamentos utilizados	41
4.3 NeTraMet	43
4.4 Metodologia dos Experimentos	44
5 Resultados	52
5.1 Objetivo Geral dos Experimentos	52
5.2 Primeiro Experimento: RIO submetido a uma carga real de usuários interativos sem a utilização de cache	53
5.2.1 Descrição e Objetivos	53
5.2.2 Descrição dos Resultados Esperados	53
5.2.3 Resultados e Análise	54

5.3	Segundo Experimento: RIO submetido a uma carga real de usuários interativos com a utilização de cache	66
5.3.1	Descrição e Objetivos	66
5.3.2	Descrição dos Resultados Esperados	71
5.3.3	Resultados e Análise	71
5.4	Terceiro Experimento: RIO submetido a uma carga real com replicação de conteúdo	75
5.4.1	Descrição e Objetivos	75
5.4.2	Descrição dos Resultados Esperados	77
5.4.3	Resultados e Análise	77
5.5	Quarto Experimento: RIO submetido a uma carga sintética	81
5.5.1	Descrição e Objetivos	81
5.5.2	Descrição dos Resultados Esperados	81
5.5.3	Resultados e Análise	81
6	Conclusões e Trabalhos Futuros	87
6.1	Trabalhos Futuros	89
	Referências Bibliográficas	90

Lista de Figuras

2.1	Arquitetura VOD	9
2.2	Arquitetura baseada em Proxies	11
2.3	<i>Rede de distribuição de conteúdo</i>	13
2.4	Técnica <i>striping</i>	17
2.5	Alocação aleatória de blocos	18
2.6	Desbalanceamento de carga a curto prazo	19
3.1	Visão simplificada do servidor RIO	26
3.2	Visão detalhada do servidor RIO	26
3.3	Fluxo Lógico do Router	29
3.4	Estrutura do Storage Server	30
3.5	Interface do RioMMClient	33
3.6	Formato típico de um log de comportamento do emulador	35
3.7	Formato típico de um log de saída do emulador	37
4.1	Arquitetura da rede Giga para os testes	39
4.2	Proposta original do projeto DIVERGE	40
4.3	Arquitetura da rede Giga na UFRJ	42

4.4	Arquitetura usada nos experimentos	49
4.5	Pontos de perda	50
5.1	<i>Goodness</i> em função do número de clientes. 1 a 5 <i>storage servers</i> , sem utilização de <i>cache</i> e sem replicação.	55
5.2	Porcentagem de <i>bytes</i> perdidos e atrasados em função do número de clientes X. 3 e 5 <i>storage servers</i> , sem utilização de <i>cache</i> e sem replicação.	56
5.3	Gráfico que exhibe, para cada configuração do servidor, o número de clientes atendidos com <i>goodness</i> médio de até 98% e 99% respectiva- mente.	57
5.4	Gráfico do tráfego gerado no experimento com 2 <i>Storages</i> e 75 clientes. Sem replicação e sem uso de <i>cache</i>	58
5.5	Gráfico do tráfego gerado no experimento com 4 <i>Storages</i> e 225 cli- entes. Sem replicação e sem uso de <i>cache</i>	58
5.6	Gráfico do tráfego gerado no experimento com 5 <i>Storages</i> e 300 cli- entes. Sem replicação e sem uso de <i>cache</i>	59
5.7	Gráficos dos experimentos com as configurações onde existem 25 cli- entes em média por <i>Storage Server</i>	62
5.8	Gráficos dos experimentos com as configurações onde existem 50 cli- entes em média por <i>Storage Server</i>	63
5.9	Gráficos dos experimentos com as configurações onde existem 75 cli- entes em média por <i>Storage Server</i>	64
5.10	Gráficos dos experimentos com as configurações onde existem 100 clientes em média por <i>Storage Server</i>	66
5.11	Histogramas gerados a partir dos experimentos com 25 clientes em média por <i>Storage Server</i> , trechos de 300 a 1800 segundos.	67

5.12	Histogramas gerados a partir dos experimentos com 50 clientes em média por <i>Storage Server</i> , trechos de 300 a 1800 segundos.	68
5.13	Histogramas gerados a partir dos experimentos com 75 clientes em média por <i>Storage Server</i> , trechos de 300 a 1800 segundos.	69
5.14	Histogramas gerados a partir dos experimentos com 100 clientes em média por <i>Storage Server</i> , trechos de 300 a 1800 segundos.	70
5.15	<i>Goodness</i> médio dos experimentos com 50 clientes por <i>storage server</i>	71
5.16	Gráfico do número de clientes X <i>Goodness</i> , 1 a 5 <i>storage servers</i> , com utilização de <i>cache</i> e sem replicação.	72
5.17	Comparação entre o <i>goodness</i> médio percebido pelos clientes com e sem a utilização de <i>cache</i>	73
5.18	Gráfico que exhibe, para cada configuração do servidor, o número de clientes atendidos com <i>goodness</i> médio de até 98% e 99%, respectivamente.	74
5.19	Gráfico que exhibe o <i>goodness</i> médio percebido pelos clientes com replicação de conteúdo no servidor, experimentos com 2 e 3 <i>storage servers</i> respectivamente.	78
5.20	Comparação entre o <i>goodness</i> médio experimentado pelos clientes com e sem a replicação de conteúdo no servidor. Experimentos com 2 e 3 <i>storage servers</i> sem a utilização de <i>cache</i> nos clientes.	79
5.21	Comparação entre o coeficiente de variação observado nos experimentos com e sem replicação de conteúdo. Experimentos com 2 e 3 <i>storage servers</i> sem a utilização de <i>cache</i> nos clientes	80
5.22	Gráfico do número de clientes X <i>Goodness</i> . 1 a 4 Clientes, sem utilização de <i>cache</i> , sem replicação e com utilização de carga sintética.	82
5.23	Comparação do <i>goodness</i> observado entre os experimentos que utilizam carga real e carga sintética.	84

5.24	Comparação entre os coeficientes de variação obtidos nos experimentos com carga real e carga sintética.	85
------	---	----

Lista de Tabelas

4.1	Relação do <i>hardware</i> das máquinas servidoras.	41
4.2	Distribuição dos Storage Servers	44
4.3	Comparação dos atrasos medidos na Rede Giga e na Internet, em milissegundos.	48
5.1	Média e Desvio Padrão nas taxas geradas pelo Storage Server sem a utilização de <i>cache</i> no cliente e sem replicação, MBits/Seg. Amostras coletadas entre 300 e 1800 segundos de duração do experimento. . . .	60
5.2	Coefficiente de variação observados em experimentos com a mesma média de clientes por <i>Storage Server</i>	61
5.3	Taxa média e limites sob qual se localizam 95% das amostras coletadas (KBits/seg). Experimentos com 25, 50, 75 e 100 clientes por <i>Storage Server</i>	65
5.4	Média e Desvio Padrão das taxas geradas pelo Storage Server com a utilização de <i>cache no cliente</i> , MBits/Seg. Amostras coletadas entre 300 e 1800 segundos de duração do experimento.	75
5.5	Economia de banda com a utilização de <i>cache</i> em relação aos experimentos sem utilização de <i>cache</i>	76
5.6	Comparação entre o coeficiente de variação das taxas em experimentos sem e com utilização de <i>cache</i>	76

5.7 Média e Desvio Padrão das taxas geradas pelo Storage Server com replicação de conteúdo, MBits/Seg. Amostras coletadas entre 300 e 1800 segundos de duração do experimento. 78

5.8 Média e Desvio Padrão das taxas geradas pelo Storage Server quando submetido a uma carga sintética. Amostras coletadas entre 300 e 1800 segundos de duração do experimento. 83

Glossário de Redes

- ACK : Mensagem de Confirmação (*Acknowledge*).
- Canal : Meio através do qual trafegam os pacotes (*Link*).
- Codec : Codificador/decodificador de sinais de áudio ou vídeo.
- FEC : Mecanismo de Correção de Erros (*Forward Error Correction*).
- Internet : A rede de computadores mais popular atualmente, composta por um conjunto de canais (*links*), computadores (*hosts*) e roteadores (*routers*), prestando serviços como correio eletrônico (*e-mail*), web e suporte a video-conferências, dentre outros, para milhões de usuários ao redor do mundo.
- Jitter : A medida da variação do atraso em pacotes enviados sucessivamente por uma rede.
- MOS : Nota de opinião média – um indicador subjetivo de QoS (*mean opinion score*).
- QoS : Qualidade de Serviço (*Quality of Service*).
- Roteador : Um dispositivo que recebe mensagens e as encaminha para seus destinos, procurando selecionar a melhor rota disponível.
- RTP : Protocolo de Tempo-Real (*Real-Time Protocol*).
- RTT : Tempo para um pacote trafegar da origem ao destino, e voltar do destino para a origem (*Round Trip Time*).
- Taxa de recepção : Taxa, em bits por segundo, com a qual os dados são recebidos por um computador na rede.

- TCP : Protocolo de Controle de Transmissão (*Transmission Control Protocol*). O protocolo de transmissão de dados mais utilizado na Internet, que oferece garantia de entrega dos dados, controle de congestionamento e controle de fluxo.
- UDP : Protocolo de Datagrama do Usuário (*User Datagram Protocol*). Protocolo de transmissão de dados minimalista, que não oferece garantia de entrega dos dados, controle de congestionamento ou controle de fluxo. É usado primordialmente para transmissão de dados multimídia como vídeo e voz.

Capítulo 1

Introdução

A queda do custo de conexões banda larga e o aumento da diversidade de serviços oferecidos através da Rede são fatores que contribuem para consagrar a Internet como uma nova mídia de comunicação, entretenimento e educação ao alcance de uma parcela significativa da população.

Essa popularização, aliada aos avanços na tecnologia de redes e a maturidade das técnicas de codificação de objetos multimídia, possibilitou o surgimento de novos serviços relacionados à transmissão multimídia via Internet, tais como ensino a distância, aplicações de voz, visualização de vídeos, videoconferência, entre outras. Dois exemplos desse tipo de serviço são o YouTube[17], página que permite aos usuários o compartilhamento de vídeos, e o Skype[16], que permite a comunicação via Internet utilizando voz e vídeo.

Dos serviços citados, vale um destaque para o ensino a distância por estar intimamente relacionado à proposta deste trabalho e por permitir a disseminação do conhecimento de maneira mais democrática, atingindo uma parcela da população que, por diversos fatores, não têm acesso aos métodos de ensino convencionais.

Devido a sua natureza, a transmissão de conteúdo multimídia possui requisitos de qualidade de serviço (*QoS*) muito diferentes do conteúdo *web* tradicional (texto e imagem). Para transmissão de conteúdo multimídia, seria ideal um tratamento diferenciado entre pacotes gerados por aplicações multimídia e por aplicações *web*

tradicionais. Porém, a infraestrutura atual da Internet não oferece garantias de qualidade e nem mecanismos para diferenciação de serviços.

Um objeto multimídia consiste em uma seqüência de dados (geralmente quadros de um vídeo ou amostras de áudio) que deve ser transmitida respeitando rígidas restrições de tempo. Perdas durante a transmissão são aceitáveis, mas se ocorrerem em excesso podem comprometer a qualidade de reprodução.

Devido a essa limitação, para atingir a QoS necessária para aplicações multimídia diversos mecanismos foram desenvolvidos na camada de aplicação. Por isso, um desafio a ser superado está relacionado ao projeto de servidores que suportem esse tipo de aplicação, um servidor multimídia deve ser robusto o suficiente para armazenar e manipular uma grande quantidade de informação obedecendo a restrições de tempo muito rígidas.

A cada dia novos provedores de serviços multimídia surgem e o número de usuários não para de crescer. Isso torna necessário o desenvolvimento e aprimoramento de técnicas buscando atender essa demanda de maneira satisfatória. Como exemplo, em um artigo do *Wall Street Journal*[33] publicada em 2006, foi observado, em um mês, um aumento de 20% na quantidade de vídeos hospedados no Youtube.

1.1 Objetivo e Motivação

Nesse trabalho, é feito um estudo do comportamento RIO (*Randomized I/O Multimedia Storage Server*). Ele foi inicialmente desenvolvido pela UCLA (*University of California Los Angeles*) e totalmente re-projetado e aprimorado através de um programa de cooperação UFRJ/UCLA. É um servidor multimídia distribuído que tem como principal característica a alocação aleatória dos blocos de cada objeto armazenado entre os discos que compõe o sistema. Atualmente, o RIO é desenvolvido no LAND (Laboratório de Análise e Desenvolvimento de Sistemas) onde passa por um processo contínuo de aprimoramento e adição de novas funcionalidades. O RIO também vem sendo utilizado em um projeto de ensino a distância do CEDERJ.

Serão realizados vários experimentos com o servidor RIO avaliando escolhas de projeto tais como: utilização de *cache* pelos clientes, replicação de conteúdo e o posicionamento de nós de armazenamento em redes com características diferentes. O objetivo do trabalho é definir o quanto cada escolha de projeto pode impactar na qualidade do serviço prestado. A carga utilizada para a realização dos experimento foi obtida a partir da coleta de *logs* de comportamento de alunos do CEDERJ (Centro de Educação Superior a Distância do Estado do Rio de Janeiro).

O servidor multimídia RIO vai atuar distribuído entre algumas instituições de ensino, a saber: UFF (Universidade Federal Fluminense), Fiocruz, UFMG (Universidade Federal de Minas Gerais) e UFRJ (Universidade Federal do RIO de Janeiro). Os experimentos foram realizados com o servidor operando em uma rede com alta capacidade, a Rede Giga, coordenada pela RNP (Rede Nacional de Pesquisa) e o CPqD (Centro de Pesquisas e Desenvolvimento). A Rede Giga disponibiliza taxas de até 10 Gbit/s para as aplicações que a utilizam e o seu objetivo é auxiliar no desenvolvimento de projetos em redes de comunicação de dados.

A ligação entre todas as instituições de ensino é feita através da Rede Giga, exceto a UFMG que será acessada através da Internet. Cada instituição vai possuir um servidor de armazenamento, de modo que o conteúdo acessado vai estar distribuído entre os servidores localizados nessas instituições.

Esse trabalho faz parte do projeto DIVERGE (Distribuição de Vídeos em Larga Escala sobre Redes Giga, com Aplicações a Educação), uma descrição mais detalhada da proposta desse projeto será apresentada no capítulo 4.

Os *logs* de comportamento foram usados diretamente para a realização dos experimentos e também de maneira indireta, através da alimentação de um modelo capaz de gerar carga sintética para o servidor RIO. Foi necessária a utilização de um emulador desenvolvido durante um trabalho anterior [34]. Esse emulador é alimentado pelos logs de comportamento e, de maneira semelhante à um cliente comum, tem a função de enviar as requisições de blocos para o servidor e esperar pela recepção dos mesmos. Sua principal diferença é o fato de que o vídeo não é exibido na tela, dessa maneira a máquina não tem a sobrecarga inerente a decodificação do vídeo. Isso

permite que vários emuladores sejam disparados em cada máquina participante, o que é fundamental para a viabilidade dos testes. Durante o desenvolvimento desse trabalho, as funcionalidades do emulador foram estendidas para calcular a estatísticas relativas à quantidade de informação recebida pelo cliente. Essas estatísticas são importantes para avaliar a qualidade do serviço prestado aos clientes.

A utilização da Rede Giga é fundamental para a realização dos experimentos em larga escala. Permitindo também a geração de uma carga considerável bem como a participação de outras instituições de pesquisa nos experimentos. Sem a Rede Giga, a distribuição do servidor não teria resultados satisfatórios devido ao retardo de transmissão em redes tradicionais. Em um dos ambientes de teste, um dos nós do servidor vai estar funcionando na UFMG e os dados serão trocados via Internet. Esse cenário vai contribuir para aprofundar o conhecimento sobre o funcionamento do servidor RIO em ambientes heterogêneos.

Um trabalho anterior já foi desenvolvido visando estudar o desempenho do servidor em um ambiente heterogêneo [21], naquele trabalho os clientes eram caracterizados por um acesso seqüencial de curta duração ao conteúdo do servidor. O trabalho dessa dissertação se diferencia do anterior porque agora foram utilizados clientes com comportamento interativo e de longa duração e pela coleta de medidas do lado servidor.

1.2 Contribuição

Uma das contribuições deste trabalho está em uma melhor compreensão do comportamento do servidor RIO graças à experimentos em uma rede de alta velocidade, usando uma carga real, coletada de um projeto de ensino distância, que é caracterizada por clientes com um comportamento interativo. Os experimentos possibilitaram:

- A avaliação da escalabilidade do servidor;
- A avaliação de estratégias de uso de *cache* no cliente;

- A avaliação de uso de replicação de blocos no servidor;
- A avaliação do tráfego enviado pelos servidores de armazenamento;
- O estudo da acurácia de um modelo para geração de carga sintética.

Outras contribuições importantes foram:

- A implantação de uma infra-estrutura para a realização de experimentos em larga-escala, utilizando a rede giga. Que agora está disponível para ser utilizada em trabalhos futuros;
- A adição de novas funcionalidades ao emulador de clientes, de maneira que agora este pode ser utilizado como uma ferramenta para avaliação da qualidade do serviço prestado pelo servidor;
- A detecção e correção de problemas no código do cliente e do servidor RIO.

1.3 Organização do trabalho

A organização deste trabalho segue o esquema descrito a seguir: No capítulo 2 é feita uma revisão de propostas relacionadas a transmissão de conteúdo multimídia na Internet. No capítulo 3 é apresentada uma descrição da arquitetura do RIO. Uma descrição do ambiente onde os experimentos foram realizados pode ser encontrada no capítulo 4. Os resultados obtidos a partir dos experimentos bem como uma análise dos mesmos pode ser encontrada no capítulo 5. Por fim, as conclusões e propostas para trabalhos futuros podem ser encontrados no capítulo 6.

Capítulo 2

Aplicações Multimídia na Internet

A popularização da Internet, somada aos avanços na tecnologia de redes, à maturidade das técnicas de codificação de objetos multimídia e ao aumento do poder de processamento dos computadores, possibilitou o surgimento de aplicações de mídia contínua tais como áudio e vídeo sob demanda, bibliotecas digitais, ambientes de realidade virtual, educação a distância, entre outros. Essa demanda nem sempre pode ser atendida de maneira satisfatória já que a infra-estrutura da Internet não é a ideal para o suporte a esses tipos de aplicações.

Os objetos multimídia tais como vídeos, trechos de áudio, textos e figuras geralmente são codificados e armazenados em discos de um servidor multimídia. Devido a sua natureza, esses objetos podem apresentar uma grande variabilidade no seu tamanho de armazenamento: em um mesmo servidor podem ser encontrados objetos com apenas alguns kilobytes ou centenas de megabytes de tamanho. Para objetos multimídia, que geralmente possuem um tamanho mais elevado, a transmissão completa para o cliente, como acontece com objetos *web* tradicionais, pode provocar um atraso inaceitável para o início da sua reprodução. Visando contornar esse problema, muitos estudos têm sido realizados buscando desenvolver protocolos de *streaming*, esses tipos de protocolo permitem que o cliente decodifique e visualize o objeto multimídia, a medida que os pacotes de dados são enviados pelo provedor.

A utilização de protocolos de *streaming* impõe rígidos limites de tempo para a

recepção dos blocos no cliente, se esses limites não forem respeitados, a qualidade de exibição pode ficar comprometida. Essa característica dos protocolos de *streaming* é algo que deve ser levado em consideração durante o desenvolvimento de qualquer servidor multimídia. Outro fator complicador é o de que a maioria dos vídeos é codificada com taxas variáveis (VBR - *Variable Bit Rate*), o que torna os requisitos de tempo ainda mais complexos.

Também deve ser observada a alta carga imposta à rede por aplicações multimídia sob o ponto de vista de uma aplicação individual ou de um agregado das mesmas. Por este motivo, várias propostas vem sendo feitas com a intenção de utilizar os recursos da rede de maneira mais inteligente, através de técnicas tais como compartilhamento de fluxos, replicação de conteúdo próximo aos clientes, troca de conteúdo entre clientes, entre outras.

As aplicações multimídia podem ser classificadas de acordo com vários critérios, a seguir serão apresentados alguns desses critérios e as classificações possíveis a partir de então.

2.1 Classificações quanto a natureza do serviço prestado

Em [40], as aplicações multimídia são divididas de acordo com a natureza do serviço prestado, assim temos as seguintes classes.

2.1.1 Transmissão de Áudio e Vídeo Pré-Armazenados

Nessa classe de serviço, o conteúdo já foi previamente codificado e armazenado no servidor. O cliente então se conecta ao servidor e pode executar comandos como pausa, avanço e retrocesso. Isso é feito através do envio de requisições ao servidor que em resposta envia os blocos solicitados pelo cliente.

2.1.2 Transmissão de Áudio e Vídeo ao Vivo

Nessa classe de serviço, o conteúdo é codificado e transmitido ao mesmo tempo em que está sendo gerado. Por ser um conteúdo que está sendo gerado dinamicamente, o cliente tem um nível de interatividade limitado, só é possível fazer uma movimentação pelo conteúdo que já foi recebido caso este tenha sido armazenado.

2.1.3 Transmissão de Áudio e Vídeo Interativos em Tempo Real

Essa classe de serviço possibilita a comunicação entre diversos clientes utilizando áudio e vídeo. O conteúdo é codificado e transmitido ao mesmo tempo que é gerado. Devido a sua natureza, esse serviço possui as restrições de tempo muito estreitas já que, para evitar desconforto dos clientes, o tempo de envio dos pacotes tem que ser da ordem de milissegundos.

2.2 Classificações quanto a distribuição dos componentes

Os servidores multimídia também podem ser classificados de acordo com a distribuição e a função de cada componente participante do sistema.

2.2.1 VoD

A arquitetura mais simples para a transmissão de um fluxo multimídia é composta por um único servidor. Esse servidor será responsável pela indexação do conteúdo multimídia, atendimento das requisições dos clientes e envio dos dados. Esse tipo de arquitetura pode ser visto em trabalhos como [20] e [57] e está ilustrado na Figura 2.1.

Uma das limitações apresentadas por essa arquitetura diz respeito à possuir

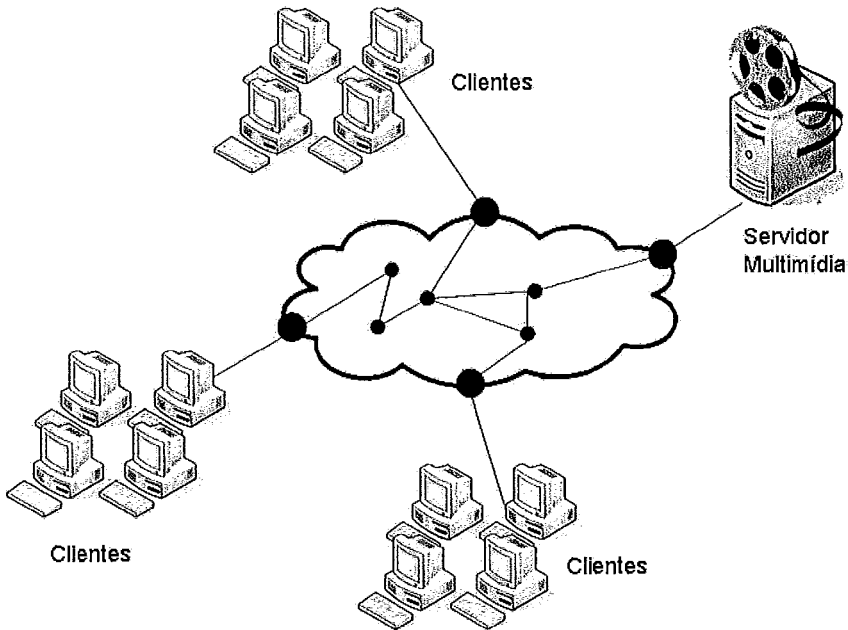


Figura 2.1: Arquitetura VOD

um único ponto de falha. Se o servidor multimídia falhar, todo o sistema ficará comprometido.

Para atender às necessidades de largura de banda e capacidade de armazenamento, um servidor desse tipo pode possuir centenas ou até milhares de discos rígidos. O tempo médio de falha em um sistema com essa proporção de discos pode ser muito baixo, comprometendo a disponibilidade do sistema [20]. Outra limitação relevante diz respeito à escalabilidade desse tipo de sistema. A transmissão de objetos multimídia pode necessitar de uma largura de banda de alguns Gbps (gigabits por segundo). Devido aos custos de transmissão, pode ser economicamente inviável atender à uma quantidade elevada de clientes com uma QoS aceitável utilizando um sistema baseado em uma arquitetura deste tipo [26].

2.2.2 DVoD

Em um servidor multimídia tradicional, o dispositivo onde os objetos multimídia estão armazenados pode possuir uma largura de banda limitada e geralmente está distante dos clientes em potencial. Visando atender um número maior de usuários

e diminuir os a largura de banda necessária é comum a divisão da responsabilidade pelo armazenamento e transmissão dos objetos entre vários servidores localizados próximos aos clientes. Esse tipo de arquitetura é chamada de servidor de vídeo distribuído ou DVoD (*Distributed Video on Demand*).

A principal vantagem desta técnica é o fato de que o custo de transmissão entre os servidores distribuídos e os clientes ser muito pequeno em comparação ao custo de transmissão a partir de um servidor central, o que possibilita um aumento na escalabilidade do sistema. Em [19] foi feito uma análise comparativa entre arquiteturas centralizadas e distribuídas.

Nesta técnica, os servidores podem funcionar como *proxies*, armazenando dinamicamente partes do conteúdo de um repositório central, de acordo com a requisições dos clientes mais próximos[22]. Também é possível que não exista um repositório central. Dessa maneira, os vídeos são distribuídos entre todos os servidores do sistema e podem ser replicados ou não. Esse é o caso do servidor RIO que será detalhado no capítulo 3.

Proxies

Uma das propostas de arquitetura DVoD faz uso de um repositório central onde os vídeos são armazenados e vários servidores distribuídos (*proxies*) posicionados próximos aos clientes em potencial. Esses servidores funcionam replicando dinamicamente o conteúdo do servidor central, de acordo com as requisições dos clientes. Dessa maneira, existe uma economia da largura de banda requerida entre o repositório central e os clientes em troca de um requerimento maior de capacidade de armazenamento. Essa troca geralmente é lucrativa, devido ao baixo custo dos dispositivos de armazenamento. Sendo assim, esses *proxies* devem armazenar o conteúdo que é acessado com mais frequência e devem possuir uma capacidade de armazenamento e largura de banda suficiente para aliviar de maneira satisfatória a carga imposta sobre o repositório central. A Figura 2.2 representa uma arquitetura baseada em *proxies*.

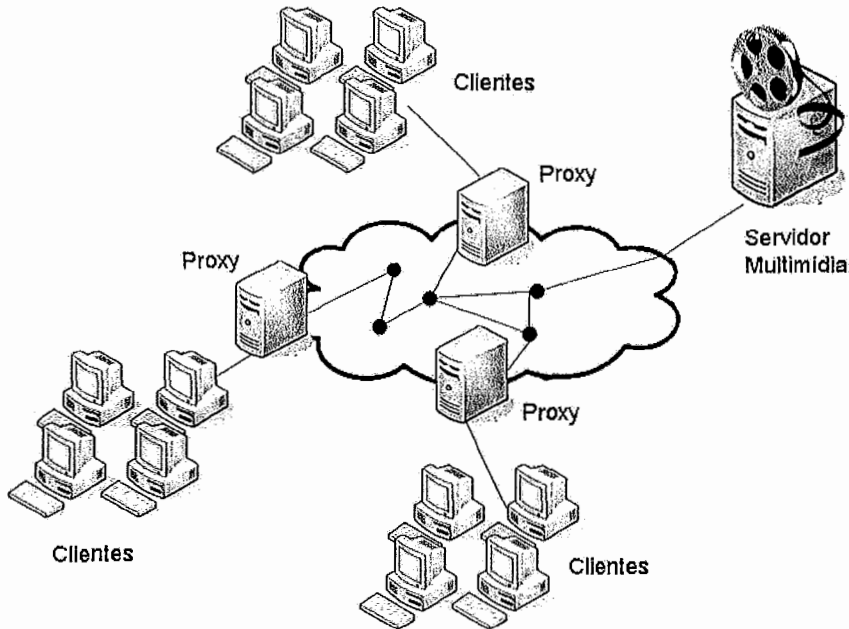


Figura 2.2: Arquitetura baseada em Proxies

O custo/benefício de uma arquitetura que usa *proxies* é extremamente dependente de fatores que devem ser levados em conta durante o projeto do sistema. Alguns fatores mais importantes são: o meio de transmissão do repositório central para os *proxies* (*unicast* ou *multicast*), possibilidade de trocas de conteúdo entre *proxies*, uso de técnicas de compartilhamento de banda, trecho do vídeo a ser armazenado no *proxy* (prefixo ou vídeo inteiro) e etc.

Existem vários estudos na literatura tentando definir qual a melhor escolha ao se desenvolver esse tipo de sistema, em [22], é feito um estudo de algumas variações na arquitetura: são estudados casos onde o repositório pode transmitir dados via *multicast* ou somente *unicast*, também são estudados casos onde os *proxies* podem trocar o conteúdo entre si. Algumas discussões a respeito de quais trechos de cada vídeo devem ser armazenados no *proxy* também são feitas.

Em [18], é feito um estudo sobre o melhor conteúdo a ser armazenado em cada *proxy* em função da capacidade de transmissão *multicast* entre o repositório central e os *proxies*. Naquele trabalho também são feitos estudos com a utilização do protocolo *badwidth skimming* para compartilhamento de fluxos.

