

Capítulo 11

Servidores Multimídia de Vídeo

Neste capítulo serão apresentadas algumas aplicações multimídia cujo objetivo não é a comunicação entre pessoas, mas a comunicação pessoa-a-sistema. Desde que estes sistemas baseados em computadores podem servir a uma variedade de funções, eles são chamados de servidores.

11.1 Comunicação Assíncrona e Síncrona

Duas formas de transmissão são possíveis nesta classe de aplicação multimídia:

- **Transmissão Assíncrona ou telecarga, armazenamento e apresentação:** neste modo a informação, ou parte dela, é primeiro totalmente transferida e armazenada no receptor, para depois ser apresentada. A informação pode ser pedida pelo usuário ou espontaneamente distribuída pelo servidor. Este modo é mais simples de implementar que o modo de transmissão síncrona.
- **Transmissão Síncrona ou tempo-real:** neste modo parte ou toda a informação é transferida em tempo-real sobre a rede e apresentada continuamente no receptor. Este modo de comunicação exige um suporte de comunicação tempo-real.

Tomamos como exemplo uma enciclopédia eletrônica para entender que modo de sincronização utilizar na transferência de informações multimídia. Uma enciclopédia contém uma composição de textos, imagens, áudios e vídeos. É fora de questão telecarregar completamente uma enciclopédia no sistema cliente. Muitos sistemas cliente poderiam não ter a capacidade de armazenamento necessária. Assim, a enciclopédia é formada por partes que o usuário pode navegar através, e que podem ser transferidas independentemente.

Parte contendo texto apenas pode ser transferida no modo assíncrono ou síncrono. Como textos não exigem muito da rede, o usuário praticamente não notará diferenças entre estes dois modos.

Para mídias contínuas do tipo áudio e vídeo, geralmente na leitura da enciclopédia o usuário é convidado, via uma interface gráfica, para ativar estas seqüências clicando um botão. Na teoria, seguido o clique do botão a seqüência de áudio ou vídeo pode ser transferida no modo síncrono ou assíncrono:

- **Modo assíncrono:** é usado normalmente para pequenas seqüências que podem ser facilmente armazenadas no receptor (mesmo na memória central do sistema cliente). Neste caso a apresentação será realizada após a carga completa (Figura 80a). Geralmente, atrasos de 3 a 4 segundos no início da apresentação são toleráveis. Para seqüências pequenas e boas condições de rede, o usuário terá a impressão de uma reação em tempo-real do servidor.
- **Modo síncrono:** ele é necessário para seqüências de áudio e vídeo muito grandes, redes muito lentas, ou pouca capacidade de armazenamento no sistema cliente. Neste caso, o sistema não aguarda a carga completa da seqüência (Figura 80b), há um pequeno atraso inicial e em seguida a seqüência é apresentada na medida que pacotes chegam.

Muitos sistemas não suportam o modo de transferência síncrona. Como resultado, o conteúdo do documento deve ser estruturado de modo que longas seqüências são cortadas em partes menores.

Problemas na Transmissão Síncrona (Tempo-Real)

Neste capítulo, nós vimos que as informações multimídia transferidas podem ser compostas por mídias discretas (textos, gráficos), mídias contínuas (som e vídeos) e relações temporais podem existir entre estes elementos. Quando transferida sobre a rede as relações temporais podem ser alteradas devido a variação do atraso de trânsito de algumas redes, onde certos fragmentos podem ser atrasados em relação a outros. Além disso, certos fragmentos podem ser perdidos ou a ordem pode ser alterada.

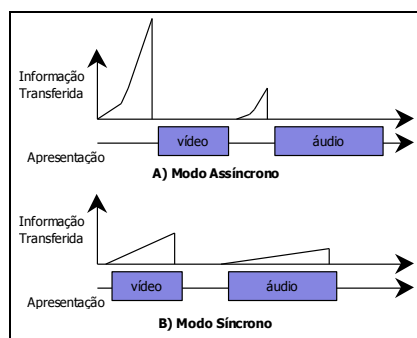


Figura 80. Modos de transmissão

Se alterações temporais ocorrerem, é necessário utilizar mecanismos no sistema cliente e servidor a fim de:

- restaurar as relações temporais dentro de cada fluxo transportando uma mídia contínua. Este processo é chamado de Streaming ou sincronização intra-mídia, que assegura que o fluxo recebido seja rearranjado para se parecer com o original. É necessário, por exemplo, garantir as dependências temporais intra-mídia entre os quadros de um vídeo.
- restaurar relações temporais entre os vários fluxos ou elementos (Sincronização Intermídia). Em outras palavras, restaurar dependências temporais inter-mídia.

11.2 RTSP (Real-Time Streaming Protocol)

O RTSP (*Real Time Streaming Protocol*) (RFC 2326) é uma proposta de padrão IETF (*Internet Engineering Task Force*) para o controle de fluxo de mídia sobre redes IP (incluindo Internet). Ele foi submetido juntamente para a IETF em outubro de 1996 pela RealNetworks e Netscape Communications Corporation e com o suporte de 40 companhias que trabalham em multimídia. O draft foi desenvolvido pela RealNetworks, Netscape, Columbia University, e o Grupo de trabalho IETF MMUSIC, e foi publicado como uma proposta de padrão IETF em Abril de 1998 [Schulzrinne, 98].

O RTSP é um protocolo cliente-servidor a nível de aplicação que permite o controle de fluxos multimídia transferidos, por exemplo, via RTP e outros (UDP, *multicast* UDP, TCP). Ele fornece métodos para realizar comandos similares às funcionalidades providas por um leitor de CD ou um videocassete, tal como partir apresentação, avanço rápido, pausa, parada e registrar. Em outras palavras, RTP age como um "controle remoto de rede" para servidores multimídia. A fonte do dado pode incluir dados ao vivo ou armazenados.

RTSP compartilha muitas propriedades com o protocolo HTTP (usado atualmente na Web). A motivação é para reutilizar a tecnologia que foi desenvolvida para o HTTP (*caching* de conteúdo, autenticação, criptografia, PICS - Plataforma para seleção de conteúdo Internet) quando acessando conteúdos multimídia tempo-real. Outras informações sobre este protocolo pode ser encontrado no URL: <http://www.cs.columbia.edu/~hgs/rtsp/>.

11.3 Vídeo sob-demanda (VOD)

Vídeo sob demanda cobre todas as aplicações onde os usuários possam pedir acesso a servidores de imagens fixas e animadas numa base individual. Verdadeiros VOD são extremamente custosos em termos de poder para acessar e ler o dispositivo de armazenamento, poder de processamento nos servidores e consumo de largura de banda na rede. Imagine que em uma hora de pico, pode haver vários pedidos de um vídeo popular ao mesmo tempo, separados por intervalos de segundos. Verdadeiros VOD gerariam em um dado instante centenas de fluxos idênticos defasados apenas de poucos segundos. Diz-se que cada cliente tem uma comutação de fase independente no mesmo filme. Q-VOD agrupa pedidos idênticos que são servidos em intervalos regulares, isto a fim de reduzir a comutação de fase e reduzir a carga do servidor. Esta técnica pode também reduzir a largura de banda, através do uso de técnicas *multicast*.

Qualidade necessária e taxa de bits associada

Uma qualidade equivalente ao videocassete VHS requer uma taxa de transmissão na ordem de 1,5 Mbits/s. De qualquer maneira, uma qualidade inferior aquela do NTSC e PAL/SECAM pode não ser aceita facilmente pelos clientes potenciais. Implementações existentes do padrão MPEG-2 opera em 6 Mbits/s com um qualidade um pouco superior ao de TV. Muitas operadoras estão planejando operar em uma infra-estrutura piloto sob uma taxa de 6 ou 7 Mbits/s.

11.3.1 Servidores VoD

Servidores multimídia são componentes importantes em aplicações baseadas em servidores do tipo mídia sob-demanda. Este capítulo apresenta alguns requisitos para este tipo de sistema, e em particular nós consideremos em detalhes os servidores de vídeo (que armazenam vídeos e áudios associados).

Um sistema servidor multimídia consiste de clientes e servidores conectados a uma rede de alta velocidade. Além disso, meta-servidores podem também ser parte de um cenário de servidor multimídia [Bernhardt, 95]. A Figura 81 mostra um modelo simplificado de um sistema multimídia distribuído em que vários clientes e servidores são interconectados por uma rede de alta velocidade. O cliente apresenta uma interface ao usuário. Pedidos dos usuários são enviados para servidores apropriados pelo cliente. Quando os servidores recebem estes pedidos, eles recuperam o dado pedido do seu dispositivo de armazenamento e os enviam para os clientes para serem apresentados aos usuários.

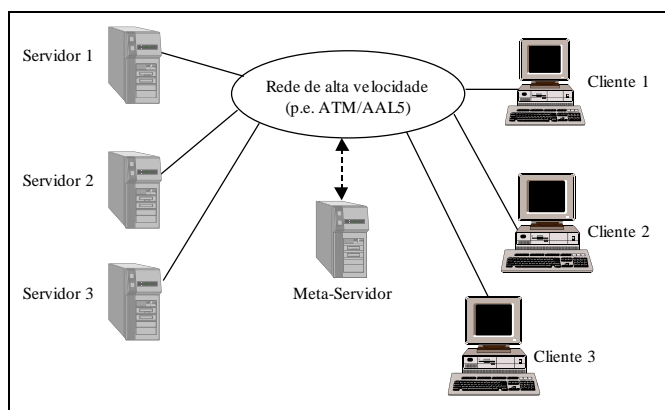


Figura 81. Arquitetura de Servidor de Vídeo

O meta-servidor fornece funções para os clientes. Por exemplo, um cliente pode perguntar ao meta-servidor os nomes e endereços dos servidores necessários para obter o fluxo de vídeo. O meta-servidor pode também fornecer informações para o cliente tal como nome de arquivos, tamanho de arquivos, taxa de quadros, esquema de compressão ou descrição do conteúdo do vídeo. Dependendo da informação recebida do meta-servidor, o cliente seleciona um servidor de vídeo apropriado.

O meta-servidor fornece funções de controle para gerenciar o sistema servidor de vídeo global. Ele pode configurar os servidores e gerenciar o sistema de armazenamento, por exemplo, vídeos populares poderiam ter cópias armazenadas em vários servidores. O meta-servidor pode também fornecer funções de controle de admissão e coletar dados necessários para faturamento, tal como o número e a duração de acesso a vídeos. Estatísticas de acesso colecionadas pelo meta-servidor podem ser usadas para otimizar o desempenho do sistema global.

11.3.2 Separação do controle e transferência de dados

Em aplicações de mídia sob-demanda, os requisitos de comunicação para mensagens de controle e transferência de mídias contínuas são diferentes. Dados de vídeo e áudio provavelmente consumirão mais largura de banda que informações de controle, mas a confiabilidade requerida para dados de controle é muito maior que para dados de vídeo. Além disso, dados de controle são trocados bidirecionalmente, enquanto fluxos de vídeo são em uma direção apenas (do servidor ao cliente).

Portanto, faz sentido usar diferentes sistemas de comunicação para trocar informações de controle e dados de vídeo e áudio [Kuo, 98]. Por exemplo, como ilustrado na Figura 82, redes ATM poderiam ser usadas para transferir dados de vídeo, enquanto ligações de baixa velocidade tal como ISDN ou linhas telefônicas fornecem capacidades suficientes para dados de controle. Para transferência de vídeo, um

serviço AAL5/ATM não confiável pode ser tolerável na medida que a perda de pacotes é abaixo de um certo limite, mas confiabilidade de 100%, como provida por um serviço TCP/IP, é necessária para troca de informações de controle. A abordagem de servidor de vídeo descrita em [IBM, 97] usa esta abordagem: AAL5/ATM para transferência de vídeo e TCP/IP para informações de controle.

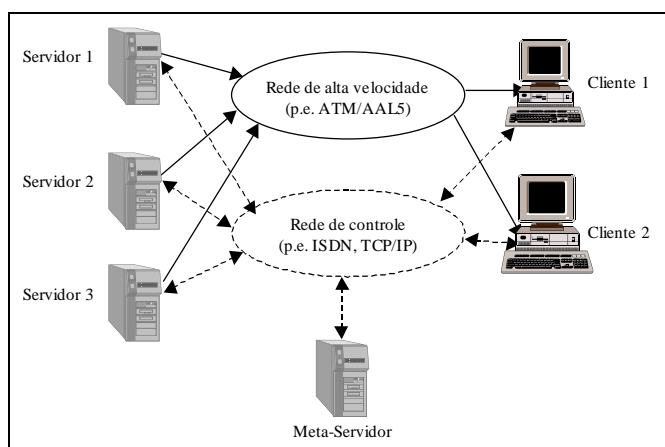


Figura 82. Redes Distintas para controle e mídias contínuas

11.3.3 Requisitos de Servidores Multimídia

Abaixo são listados alguns requisitos de servidores multimídia identificados por [Lu, 96]. Estes requisitos são impostos pelas características dos dados multimídia que são basicamente um grande volume de informações e tempo-real.

- A capacidade de armazenamento e a taxa de transferência de dados deveriam ser suficientemente alta para suportar vários clientes simultaneamente afim de tornar o sistema econômico. Vídeo e áudio digital são normalmente compactados para reduzir os requisitos de armazenamento e largura de banda de transferência. Mas mesmo após compressão, estes dados continuam a exigir uma capacidade de armazenamento e largura de banda consideráveis.
- Servidores multimídia deveriam fornecer garantias de QoS para os fluxos multimídia. Para satisfazer isto, o servidor deveria implementar controle de admissão e escalonamento tempo-real. Para fluxos a taxa de bits constante, os requisitos de QoS são relativamente fáceis de especificar, e assim o controle de admissão e escalonamento podem ser implementados com uma relativa facilidade para satisfazer os requisitos. Agora quando o fluxo é a taxa de bits variável, pode-se reservar recursos para a taxa de bits de pico para uma garantia a 100% (hard). Como visto no capítulo 7, isto levaria a um baixo uso dos recursos do servidor. Para aumentar o uso dos recursos, garantias estatísticas ou soft deveriam ser usadas e os clientes deveriam permitir um escalamento da mídia ou degradação suave da qualidade. O problema com a garantia estatística por degradação suave da qualidade é que a especificação da QoS, o gerenciamento dos recursos e o projeto da aplicação se tornam complicados.
- A arquitetura e técnica usada em um servidor deveria ser escalável e capaz de suportar uma grande população de usuários.
- Muitas aplicações multimídia são interativas, assim o servidor deveria ser capaz de suportar vários tipos de interações com o usuário tal como pausa, avanço e retrocesso rápidos.
- O servidor deveria fornecer capacidades de busca.

Modelo de Servidor Multimídia

A Figura 83 apresenta o modelo de servidor multimídia proposto por [Lu, 96]. Os principais componentes são os dispositivos de armazenamento, um escalonador e um buffer de suavização. O escalonador determina qual fluxo é o próximo a ser servido (caso existam vários fluxos sendo transmitidos simultaneamente, ou seja existam vários pedidos de vídeo que estão sendo atendidos pelo servidor). Como a leitura de dados dos dispositivos de armazenamento é realizada em rajada para fluxos individuais, buffers de suavização são usados para transmitir fluxos de mídias contínuas para aplicações (no caso de o servidor e o cliente estando em uma mesma máquina) ou para um sistema de transporte. Desta forma, o dado será transmitido para a rede na mesma taxa que ele deve ser apresentado no

cliente. A meta do servidor multimídia é servir simultaneamente tantos fluxos quanto possível satisfazendo os requisitos de continuidade dos fluxos e guardando os requisitos de bufferização ao mínimo. Para obter isto, dispositivos de armazenamento apropriados, posicionamento dos dados nestes dispositivos e técnicas de escalonamento devem ser usados.

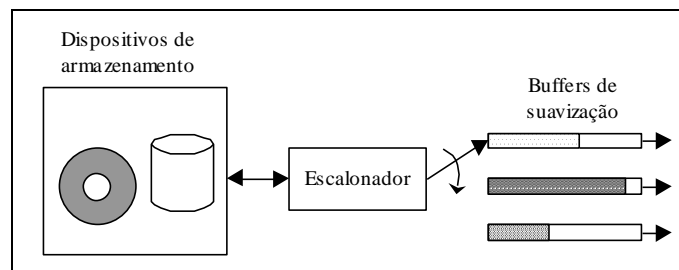


Figura 83. Um modelo de servidor multimídia [Lu, 96]

Alguns pesquisadores propõem esquemas onde um cliente obtém e armazena um arquivo multimídia completo antes da apresentação do arquivo. Estes esquemas têm a vantagem de que eles não impõem requisitos temporais no servidor e na rede. Mas eles impõem altos requisitos de armazenamento no cliente e o tempo de resposta do sistema também é muito longo quando o arquivo é grande. Além disso, o usuário pode descobrir que o arquivo transferido não é aquele que ele desejava no momento da apresentação, assim o tempo e largura de banda usados para transferir o arquivo completo é perdido.

Reserva de Recursos em Servidores Multimídia

Aplicações de servidor multimídia e servidores de vídeo em particular são normalmente baseadas em reserva de largura de banda afim de fornecer um filme com boa qualidade. Recursos podem ser reservados em avanço, definindo o intervalo de tempo desejado para apresentação. O meta-servidor então calcula o melhor tempo de partida de uma transmissão de vídeo considerando todos os outros pedidos conhecidos e seus requisitos de largura de banda.

O servidor de vídeo tem alguns conhecimentos a priori acerca do vídeo armazenado. Estes conhecimentos podem ser usados para gerenciar eficientemente os recursos de rede necessários, sendo útil para reserva de recursos para transmissão de vídeo compactado em rajadas. Como já visto neste curso, neste tipo de tráfego a reserva da taxa de pico pode desperdiçar largura de banda desde que a taxa de pico é requerida apenas para algumas poucas cenas de um filme. Uma abordagem para evitar desperdício de largura de banda é dividir o fluxo de vídeo em diferentes cenas com diferentes valores de pico de taxa de bits. Para cada cena, uma nova reserva é realizada.

Multicast otimiza uso dos recursos

Se um filme popular é pedido por vários clientes simultaneamente, estes clientes podem ser juntar a um único grupo multicast. O servidor de vídeo pode então fazer o multicast do fluxo de vídeo sob a rede de alta velocidade e usar todas as vantagens da transmissão multicast.

Nas próximas seções, nós discutiremos algumas questões e técnicas para satisfazer os requisitos apresentados nesta seção.

11.3.4 Dispositivos de armazenamento

Nesta seção veremos os requisitos de armazenamento e largura de banda para um servidor multimídia de propósito geral. Serão analisados os vários tipos de dispositivos de armazenamento e as técnicas para aumentar a capacidade de armazenamento e largura de banda de transferência.

Requisitos de capacidade de armazenamento e Largura de banda

Assuma que um servidor armazena 2000 filmes de 100 minutos cada. Estes filmes são compactados em MPEG-2 a 8 Mbps com qualidade de TV digital, e o servidor deveria servir 1000 clientes simultaneamente. Então a capacidade de armazenamento requerida é 12 TB e a taxa de transferência de 8 Gbps. Estes números indicam grosseiramente quanto um dispositivo de armazenamento ou uma combinação de dispositivos de armazenamento de um servidor típico deveria fornecer como capacidade de armazenamento e largura de banda de transferência. Note que a largura de banda de transferência é a largura de banda percebida pelos usuários e não a largura de banda de pico do dispositivo de

armazenamento. A largura de banda real pode se aproximar a largura de banda de pico apenas quando o tempo de acesso⁹ é minimizado. Minimização do tempo de acesso pode ser obtida por um posicionamento de dados e escalonamento de disco apropriados, como discutido a seguir.

Tipos de dispositivos de armazenamento

Os dispositivos de armazenamento mais comuns são discos magnéticos, discos óticos e tapes (por exemplo, fitas DAT¹⁰). Características típicas destes dispositivos são resumidas na tabela abaixo [Lu, 96]. Um jukebox¹¹ consistindo de vários discos óticos ou tapes pode fornecer maiores capacidades.

Característica	Disco magnético	Disco ótico ¹²	Low-end tape	High-end tape
Capacidade	Baixa	Média	Alta	Mais alta
Modo de acesso	Randômico, menor tempo	Randômico, maior tempo	Sequencial	Sequencial
Taxa de transf.	Mais alta	Alta	Baixa	média
Custo por GB	Mais alto	Médio	Baixo	Mais baixo

Discos magnéticos têm as características desejáveis para suportar aplicações multimídia: eles permitem um acesso randômico rápido e tem altas taxas de transferência. Mas eles são relativamente caros comparado a outros dispositivos de armazenamento. Discos óticos têm mais alta capacidade que discos magnéticos e também permitem o acesso randômico, mas o tempo de acesso é grande e a taxa de transferência é baixa. Tapes têm a mais alta capacidade de armazenamento, mas não podem ser acessados randômicamente (para ler um bloco de dados particular, é necessário ler todos os blocos precedentes) e a taxa de transferência é baixa.

A partir das características dos vários dispositivos de armazenamento apresentados na tabela acima, pode-se ver que não é possível para um dispositivo único satisfazer a capacidade de armazenamento e largura de banda de transferência requerida descrita na seção 11.3.1. A solução é usar vários dispositivos de armazenamento em um servidor. Nós descrevemos três abordagens na sequência:

- Uso de vários discos para formar um vetor de disco para aumentar a capacidade de armazenamento e largura de banda de transferência.
- Junção de vários dispositivos de armazenamento para formar uma hierarquia de armazenamento de baixo custo para formar um servidor multimídia de grande capacidade.
- Utilizar vários servidores para formar um vetor de servidores.

Vetores de Disco e RAID

Como visto anteriormente, o disco magnético é o tipo de dispositivo de armazenamento mais interessante para aplicações multimídia. Mas um simples disco não é capaz de armazenar muitos arquivos multimídia e suportar vários usuários simultaneamente. Uma solução natural é usar vários discos para formar um vetor. Formando vetores de disco aumenta-se a capacidade de armazenamento e a taxa de transferência. O aumento da taxa de transferência é possível pois um arquivo tem seus dados espalhados em diversos discos rígidos e estes dados podem ser lidos de uma maneira paralela, balanceando a carga entre diversos drives.

A tecnologia de vetor de disco muito usada em produtos comerciais e prototipos de pesquisa são os RAIDs (*Redundant Array of Inexpensive Disk*). Hoje, RAID são mais referenciados como Vetor Redundante de Discos Independentes (*Redundant Array of Independents Disk*). RAID são redundantes pois discos extra são usados para armazenar informações redundantes que são usadas para recuperar a informação original quando um disco falha. Comparado com discos grandes e caros, os RAIDs têm melhor desempenho, confiabilidade, menor consumo e escalabilidade, eles são muito usados como dispositivo de armazenamento de dados multimídia.

⁹ O tempo de acesso é o tempo necessário desde o pedido de leitura de dados no dispositivo de armazenamento ao início da transferência destes dados no dispositivo.

¹⁰ Cartuchos Digital Audio Tape (DAT) tem uma capacidade de 2 GB a 24 GB e necessitam de drives relativamente caros. Eles também têm uma taxa de transferência relativamente lenta.

¹¹ Um jukebox é um dispositivo que armazena vários CD-ROMs (até 500) e usa um braço mecânico, carrossel ou outro dispositivo para ligar o disco à estação ótica para leitura e escrita

¹² Discos óticos podem ser de três tipos: CD-ROM (somente leitura), WORM (para leitura e escrita) e Erasable optical EO (que podem sofrer escrita, serem lidos e apagados).

Hierarquia de armazenamento

Discos magnéticos são mais adequados para armazenamento multimídia mas eles são relativamente caros. Para usar a alta capacidade de discos óticos e tapes, uma hierarquia de armazenamento contendo vários tipos de dispositivos de armazenamento é usada para implementar servidores de baixo custo. Existem várias abordagens para construir tal hierarquia de armazenamento.

Uma abordagem é usar tapes e discos óticos como arquivos de armazenamento e usar discos magnéticos apenas para armazenar os segmentos iniciais dos arquivos multimídia. Estes segmentos podem reduzir a latência de inicialização e assegurar transições suaves na apresentação. Mas esta abordagem não suporta muitos fluxos ao mesmo tempo devido a taxa de transferência relativamente baixa dos discos óticos e tapes. Além disso, quando tapes são usados, o acesso randômico não é possível.

Em uma segunda abordagem, um arquivo completo é movido dos tapes para os discos magnéticos quando pedido. A deficiência desta abordagem é que o atraso inicial associado a carga do arquivo completo será muito grande para arquivos grandes tal como filmes. Este problema pode ser resolvido explorando os dois fatores seguintes:

- Primeiro, não há muitos arquivos ou filmes populares ao mesmo tempo, assim nós podemos carregar os arquivos mais populares no disco antes de eles serem pedidos ou nas suas primeiras requisições. Os discos são usados como cache para arquivos mais populares.
- Segundo, padrões de usuário são previsíveis em muitas aplicações. Por exemplo, quando um usuário requisita o episódio um de uma série, é claro que ele acessará o episódio dois quando o episódio um terminar. Assim, nós podemos carregar os arquivos previsíveis no disco antes de eles serem pedidos.

O conceito de hierarquia de armazenamento pode ser estendido para uma hierarquia de armazenamento distribuída. Uma hierarquia de armazenamento distribuída consiste de um número de níveis de servidores incluindo servidores locais, servidores remotos e arquivos remotos, como mostrado na Figura 84. Um arranjo possível destes servidores é que os servidores locais são colocados em LANs, servidores remotos em MANs ou redes de campus, e arquivos em WANs. Os servidores locais e remotos usam discos magnéticos como dispositivo de armazenamento, enquanto servidores arquivo usam discos óticos ou tapes (ou jukebox) como dispositivo de armazenamento. A vantagem desta organização é a boa escalabilidade.

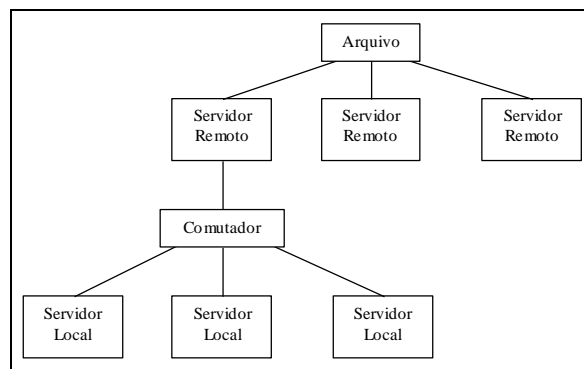


Figura 84. Uma hierarquia de armazenamento distribuída

Vetor de Servidores

Um ponto importante em sistemas servidores de vídeo é a QoS. Degradação da qualidade pode ocorrer devido a desempenho insuficiente do servidor ou da rede. Um único servidor pode normalmente não suportar um grande número de clientes com um grande número de fluxos de vídeo diferentes. Vetores de servidores são necessários. Existem duas formas de implementar vetores de servidores [Kuo, 98].

- A primeira abordagem distribui apenas filmes completos em servidores. O cliente deve perguntar ao meta-servidor o servidor de vídeo que pode ser acessado para obter o filme desejado. Esta abordagem adapta-se bem ao acréscimo do número de pedidos enquanto os servidores permanecerem descarregados e capazes de satisfazer certo número de pedidos. Um meta-

servidor deveria ter o conhecimento acerca da popularidade dos filmes e configurar um número apropriado de servidores de vídeo para armazenar os filmes mais populares.

- Outra abordagem é implementar um mecanismo de faixas (striping) [Bernhardt, 95]. Neste caso, todos os filmes são divididos em subconjuntos não sobrepostos de quadros de vídeo, e cada subconjunto de quadros é armazenado em um servidor separado. Quando um cliente pede um filme, cada servidor transmite seu subconjunto de quadros como um subfluxo. O cliente recebe os diferentes sub-fluxos e reconstrói eles em um completo fluxo de vídeo. Por exemplo, é possível armazenar todos os quadros de vídeo com os números de seqüência $3n-2$ no servidor de vídeo 1, quadros $3n-1$ no servidor 2, e quadros $3n$ no servidor 3 ($n \in \mathbb{N}$). A abordagem tem a vantagem de que a carga do servidor é bem distribuída entre os servidores e que cada servidor deve armazenar o mesmo montante de dados. De qualquer maneira, os diferentes servidores devem partir suas transmissões aproximadamente no mesmo instante, e a sincronização de fluxo deve ser realizada a fim de evitar grandes tempos de bufferização nos clientes.

11.3.5 Posicionamento de dados no disco

A meta de servidores multimídia é satisfazer restrições temporais dos fluxos multimídia e servir simultaneamente tantos fluxos quanto possível. Para obter isto, o posicionamento apropriado de dados no disco, escalonamento de disco e controle de admissão são essenciais. Esta seção trata do primeiro problema. Em muitos casos, estes três problemas são diretamente relacionados e dependentes: uma certa disciplina de escalonamento de disco requer um tipo particular de posicionamento de dados e baseia-se em um algoritmo de controle de admissão particular.

Um arquivo normalmente é quebrado em um conjunto de blocos de armazenamento para leitura e escrita. Existem dois métodos gerais para posicionar estes blocos: eles são colocados continuamente em um disco ou espalhados ao redor do disco. Para aumentar o desempenho, variações destes dois métodos foram propostos. Em [Lu, 96] várias técnicas de posicionamento de dados são apresentadas.

11.3.6 Escalonamento de Disco e Controle de Admissão

Como visto anteriormente, para comunicações multimídia de qualidade é necessário que todos os subsistemas envolvidos garantam o desempenho fim-a-fim. No caso de aplicações baseadas em servidor, o servidor é apenas um destes subsistemas: ele deve transmitir o dado para a rede na mesma taxa que ele deveria ser apresentado no cliente.

Como operações de disco são indeterminísticas devido ao tempo de acesso randômico e o compartilhamento dos recursos do servidor com outras aplicações, disciplinas de escalonamento de disco são necessárias para manter a continuidade de dados multimídia contínuos. Como em outros subsistemas, garantias de desempenho podem apenas serem obtidas quando o sistema não está sobrecarregado. Portanto, um controle de admissão é necessário para prevenir a sobrecarga do sistema.

Para evitar a falta de dados (*data starvation*), o acesso a dados no disco deveria ser feita antes do tempo de transmissão do dado (assumindo que a taxa de transmissão é a mesma que a taxa de apresentação no receptor). O dado excedente deve ser bufferizado. O tamanho deste buffer não deve ser muito grande pois um buffer grande significa que o sistema é caro e o dado sofrerá grandes atrasos. Portanto, a tarefa do servidor é prevenir a falta de dados minimizando os requisitos do buffer e atraso.

Controle de admissão

Técnicas de controle de admissão são normalmente associadas às disciplinas de escalonamento de disco. O controle de admissão no servidor deveria ser baseado no seguinte critério: *a largura de banda total (LBT) de todos os fluxos pedidos deveria ser mais baixa que a taxa de transferência (T) do disco*. Na prática, LBT deveria ser muito mais baixa que T. Isto pois não há tempo de busca¹³ zero quando a cabeça do disco termina um serviço e move-se para outro lugar para servir o próximo pedido.

¹³ Tempo de busca é o tempo levado pelo braço atuador para mover o cabeçote entre trilhas. Hoje, existe aproximadamente 3000 trilhas em cada lado de um prato de 2,5 polegadas. Assim, para ir da trilha atual para a trilha contendo o dado a ler pode implicar no movimento de uma trilha ou até 2999 trilhas. O tempo de busca de uma trilha é de aproximadamente 2 ms, e o máximo tempo de busca (movimento de 2999 trilhas) consome cerca de 20ms. O tempo de busca média dos drives atuais é de 7,6 a 14 ms.

Algoritmos de escalonamento de disco tradicionais

Algoritmos de escalonamento de disco são usados mesmo em sistemas de arquivo tradicionais, mas seu propósito é reduzir o tempo de busca e a latência de rotação¹⁴, obtendo mais alta vazão ou fornecendo acesso igual para cada aplicação. Os algoritmos de escalonamento de disco mais comuns são:

- FCFS (First-Come-First-Served), onde pedidos são servidos de acordo com sua ordem de chegada. Ele ignora o movimento e a localização do cabeçote do disco, assim o tempo médio de busca é alto.
- SSTF (Shortest Seek Time First), que tenta minimizar o tempo de busca servindo o pedido cujo dado está mais próximo da localização atual do cabeçote. SSTF favoriza o pedido no meio de um disco. Quando o servidor é muito carregado, os pedidos de transferência de dados associados às trilhas mais internas e mais externas algumas vezes não são servidos.
- Scan: também tenta minimizar o tempo de busca servindo pedidos na ordem do movimento dos cabeçotes do disco. Ele serve primeiro todos os pedidos em uma direção até que todos os pedidos sejam servidos nesta direção. O movimento dos cabeçotes são então invertido e serve os pedidos nesta direção.

Estes algoritmos de escalonamento não levam em consideração a temporização de cada fluxo. Assim, eles não podem ser diretamente utilizados para escalonamento de servidores multimídia (a não ser que seja limitado o número de pedidos).

Algoritmo EDF (Earliest Deadline First)

EDF é o algoritmo mais comum para escalonamento tempo-real de tarefas com deadlines. EDF escalona para primeiro o bloco de mídia com deadline mais próximo. A limitação do EDF aplicado ao escalonamento de disco é que ele ignora a posição do cabeçote, produzindo um grande tempo de busca e latência de rotação. Desde que um tempo não irrelevante é desperdiçado no movimento do cabeçote no disco sem realmente ler dados úteis, assim a vazão não é inteiramente usada.

Suponha que o servidor está servindo $n-1$ pedidos com garantias hard. A condição suficiente para aceitar um novo pedido n é: $(b_1 + b_2 + \dots + b_n) + (n-1) * S * r \leq r$; onde b_i é a taxa de dados máxima do fluxo i , S é o tempo de busca máximo do disco e r é a vazão do disco. Como EDF não impõe nenhum posicionamento de dados específico e o tempo de busca é indeterminístico (pode variar de 0 ao tempo de busca máximo), é difícil desenvolver um algoritmo de controle de admissão que possa satisfazer todos os requisitos de continuidade dos fluxos e ao mesmo tempo usar recursos eficientemente.

Algoritmo Scan-EDF

Scan-EDF combina Scan e EDF para reduzir o tempo de busca médio do EDF [Reddy, 94]. Scan-EDF serve o pedido com deadline mais próximo como EDF, mas quando vários pedidos tem o mesmo deadline, seus respectivos blocos são acessados com o algoritmo Scan. É claro que quanto mais pedidos tiver o mesmo deadline, mais eficiente é o Scan-EDF. Quando todos os pedidos têm um deadline diferente, Scan-EDF se reduz a EDF apenas.

Para aumentar a eficiência do Scan-EDF, algumas técnicas foram propostas para aumentar o número de pedidos com o mesmo deadline. Por exemplo, pode-se forçar que todos os pedidos tenham deadlines que são múltiplos de um período de p . Neste caso, os deadlines são servidos em blocos.

O controle de admissão do Scan-EDF é similar ao do EDF. Mas em vez de usar o tempo de busca máximo para cada chaveamento de serviço, o tempo de busca máximo é usado apenas quando do chaveamento de um pedido com um deadline para outro pedido com um deadline diferente. Assim, o scan-EDF pode servir mais pedidos simultaneamente (ou ao menos igual ao número de pedidos) que o EDF.

Algoritmo Round-Robin

No Scan, pedidos são servidos em rodadas, ou seja em cada rodada é lido parte de cada fluxo em transferência e a rodada se repete ciclicamente. Mas os pedidos são servidos em diferentes ordens em

¹⁴ Desde que o cabeçote está posicionado na trilha apropriada, ela deve aguardar para que o drive rode o prato para o setor correto. Este período de espera, chamado de latência de rotação, depende da velocidade de rotação do drive. Um drive de 3600 rpm tem uma latência de rotação média de 8,3 ms e um drive rápido de 7200 a latência média é de 4,2 ms.

diferentes rodadas dependendo da posição do dado. O intervalo de tempo máximo entre dois serviços sucessivos para um fluxo é igual ao tempo de duas rodadas. Isto pois um fluxo pode ser servido primeiro em uma rodada (uma direção de movimento do cabeçote) e o último em outra rodada (outra direção do cabeçote). No EDF, não há rodadas distintas, fluxos com pedidos freqüentes podem ser servidos várias vezes enquanto outros são servidos apenas uma vez.

No escalonamento round-robin, fluxos são servidos em rodadas distintas e a ordem de serviço de cada fluxo é fixa em cada rodada. Assim, o intervalo de tempo entre dois serviços sucessivos para um fluxo é limitado a duração da rodada. Se a duração da rodada é constante, então o intervalo também é constante. Desde que a ordem de serviço para fluxos é fixada, o posicionamento de dados é importante para determinar o tempo de busca.

Quando round-robin é usado com posicionamento de dado restrito, o tempo de busca é mínimo e a continuidade de cada fluxo é mantida. Por exemplo, na Figura 85 três fluxos podem ser obtidos continuamente. O espaçamento G para o fluxo 1 é $(M_2 + M_3)$. Durante a implementação, M e G podem ser ajustados para satisfazer o requisito de continuidade. Por exemplo, se um vídeo HDTV é compactado a 0.5 Mbits/quadro (62500 bytes/quadro) e registrado a 60 fps em um disco com uma taxa de transferência de 25600 setores/s (cada setor é igual a 512 bytes). Escolhendo que cada bloco é um bloco, então o bloco ocuparia 123 setores (62500/512). Então $M=123$. Usando a condição do posicionamento restrito ($T \geq (M+G)/r$), então $G \leq (25600/60) - 123$, ou seja $G \leq 303$ setores.

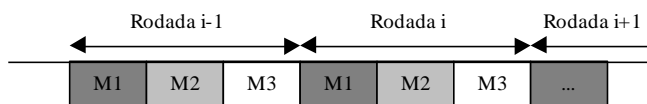


Figura 85. Posicionamento de dados restrito

O intervalo de tempo entre acessos sucessivos de um fluxo de dado tem várias implicações nos requisitos de bufferização e o atraso da partida da apresentação. Para round-robin, a apresentação de um fluxo pode ser iniciada após o acesso do primeiro bloco do fluxo (assumindo que a apresentação é na mesma máquina que o servidor). Entretanto, com Scan a apresentação deve esperar até o fim da primeira rodada (pois a posição do serviço na rodada varia). Para prevenir falta de dados no dispositivo de saída, round-robin necessita um espaço de buffer suficiente para satisfazer o consumo de dado para uma rodada, enquanto Scan necessita espaço suficiente para satisfazer o consumo de aproximadamente duas rodadas. Mas em geral, o tamanho da rodada para Scan é mais curto que aquele do round-robin como o tempo de busca é normalmente mais curto para Scan (exceto no caso de posicionamento restrito).

Escalonamento GSS (Group Sweeping Scheduling)

No escalonamento GSS, cada rodada é particionada em grupos. Cada fluxo é associado a um grupo, e os grupos são servidos em uma ordem fixa em cada rodada, tal como em round-robin. Dentro de cada grupo, Scan é usado para servir fluxos do grupo. Scan reduz o tamanho da rodada e round-robin reduz o intervalo entre serviços sucessivos. Assim GSS é um compromisso entre o tamanho da rodada e o intervalo de serviço sucessivo. A Figura 86 compara os tamanhos das rodadas e os intervalos máximos de serviços sucessivos para Scan, round-robin e GSS. O tamanho da rodada é mais curto para Scan, mas o intervalo de serviço máximo é o dobro do tamanho da rodada. O tamanho da rodada do round-robin é o mais longo, mas o intervalo de serviço máximo é igual ao tamanho da rodada. O tamanho da rodada do GSS está entre os tamanhos do Scan e round-robin. Seu intervalo de serviço máximo é um pouco maior que seu comprimento de rodada. Assim, em muitos casos, o intervalo de serviço do GSS é mais curto.

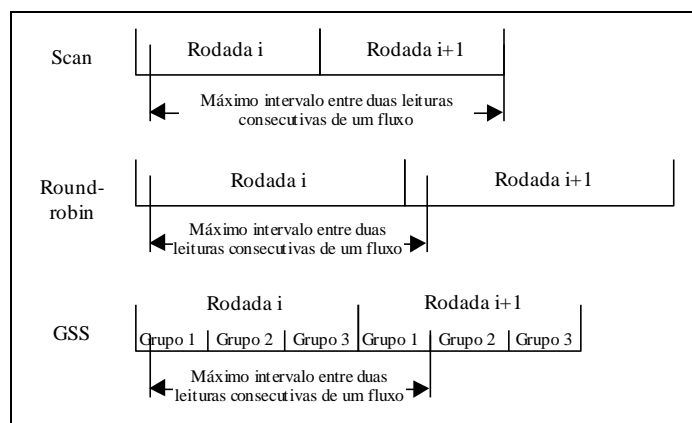


Figura 86. Comparação dos tamanhos de rodada e intervalos máximos entre duas leituras consecutivas de um fluxo para Scan, round-robin e GSS

11.3.7 Interações com o usuário

Um servidor multimídia distribuído deve fornecer uma interface para que os clientes possam pedir informações. Além disso, o sistema deveria suportar interações com o usuário tal como pausa, retomada e avanço rápido.

Um servidor pode fornecer dois tipos de interface para o cliente:

- Sistema orientado a arquivo, onde o dado multimídia é tratado como arquivos normais. O cliente utiliza operações típicas de arquivo para acessar o dado multimídia, tal como abrir, fechar e ler. O servidor pode usar a operação abrir para forçar o controle de admissão. Certa quantidade de dados é lida em cada operação de leitura. Usando esta interface, o cliente pode facilmente implementar operações de pausa simplesmente pela parada de emissão de comandos de leitura e a operação retomada pela retomada da emissão de comandos de leitura. Mas neste esquema é difícil implementar avanço e retrocesso rápidos. Além disso, é difícil manter a continuidade do fluxo usando este tipo de interface pois o comando de leitura pode sofrer atrasados indeterminados na rede.
- Sistema orientado a fluxo, onde o cliente emite um comando de partida e o servidor emite periodicamente dados em uma taxa predefinida sem pedidos adicionais de leitura. Este é o modo de operação preferido pois com este é fácil manter a continuidade do fluxo.

A seguir será descrita a implementação de operações de comando semelhantes ao videocassete, como pausa, retomada, avanço rápido e retrocesso rápido. Além destas operações, também é necessário implementar operações de busca e acesso baseado no pedido do usuário. Por exemplo, o usuário pode querer encontrar todos os vídeo-clips similares ao que ele está assistindo.

Pausa e Retomada

Se um usuário está acessando um fluxo independente de outros usuários, é relativamente fácil implementar os comandos de pausa e retomada simplesmente pela emissão dos respectivos comandos para os servidores. Mas mesmo neste caso simples, cuidados devem ser tomados para tratar possíveis dados extras enviados ao cliente após a emissão do comando pausa. Isto ocorre pois o comando pausa leva algum tempo para chegar ao servidor, e durante este tempo o servidor pode enviar dados ao cliente. Como mostra a Figura 87, o usuário emite o comando pausa no instante P, mas antes do servidor receber este comando, ele envia dados até o instante Q do fluxo. O que fazer com estes dados recebidos entre P e Q?

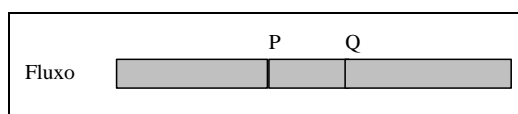


Figura 87. Comando pausa é emitido no ponto P mas o servidor para de enviar dados em Q

Existem três opções para tratar estes dados extras:

- A primeira opção é não parar a apresentação do fluxo imediatamente após o pedido de parada do usuário. Em vez disso, a apresentação continua até o ponto Q e para quando ele recebe uma confirmação de pausa do servidor. Quando o usuário emite um comando de retomada, o fluxo é retomado no ponto Q.
- Na segunda opção, o cliente para a apresentação imediatamente no pedido do comando de pausa (P) e bufferiza o dado entre P e Q. Quando o usuário emite o comando retomada, a apresentação se inicia em P usando o dado bufferizado e o servidor emite dados a partir do ponto Q.
- Uma terceira opção é parar a apresentação no instante P e descartar os dados recebidos entre P e Q. Quando o usuário emite um comando de retomada, o cliente pede ao servidor o acesso e transmissão de dados a partir do ponto P. Após um certo atraso inicial, a apresentação se inicia em P.

As opções 1 e 2 são mais eficientes pois elas usam os dados recebidos entre P e Q sem que o servidor necessite enviar novamente estes dados. Mas infelizmente é difícil obter sincronização intramídia e intermídia com estas duas opções. Isto pois o tempo entre P e Q são diferentes para fluxos diferentes devido a atrasos aleatórios na rede sofridos pela transmissão do comando pausa do cliente aos diferentes servidores. Além disso, é difícil na segunda opção tornar segura a apresentação suave no ponto Q devido a variações de atraso da rede. Portanto, a opção 3 é preferida a despeito do desperdício de largura de banda devido a retransmissão de dados entre P e Q.

Quando o servidor lê os dados uma única vez e difunde (broadcast/multicast) estes dados a vários clientes ao mesmo tempo, a implementação dos comandos de pausa e retomada é complicada devido ao compartilhamento da informação por vários clientes e é necessário que a interação de um cliente não interfira na apresentação da informação para os outros clientes. Existem duas soluções para este problema:

- Em uma primeira solução, o sistema impõe que o usuário possa apenas parar a apresentação por um certo intervalo. A abordagem para suportar tal cenário é transmitir o filme como N diferentes fluxos com tempos de partidas defasados, por exemplo, fluxo n tem um atraso de 5 minutos comparado ao fluxo n-1. Este tempo de partida defasado é chamado de diferença de fase. Esta abordagem permite ao cliente interromper a transmissão por 5 minutos. Antes da parada, o cliente recebe o fluxo n-1; após a parada ele receberá o fluxo n.
- Em uma segunda opção, um buffer grande e um gerenciamento de buffer complicado são necessários para fornecer a interação independente de clientes individuais [Wong, 95].

Avanço e retrocesso rápidos

Existem dois modos de implementar avanço rápido:

- O primeiro modo é apresentar a mídia em uma taxa mais rápida que a taxa original.
- O segundo modo é apresentar a mídia na taxa normal, mas saltando alguns quadros. Por exemplo, se quadros alternativos são saltados, o vídeo avança com uma velocidade duplicada.

A primeira opção é a mais fácil de implementar, mas ela requer grande largura de banda e poder de processamento. A rede e o cliente podem não ser capazes de satisfazerem os requisitos de largura de banda e computacional. Os requisitos podem ser reduzidos usando técnicas de codificação de vídeo escaláveis. Quando o vídeo é compactado com técnicas de subbanda (escalável), apenas a primeira banda com vídeo de baixa qualidade pode ser transmitido para operações rápidas.

O retrocesso rápido pode ser implementado de maneira similar aquelas do avanço rápido exceto que os dados são lidos para trás.

A implementação de avanço e retrocesso rápidos é complicada devido a duas características de compactação: codificação interquadro e taxa de bits variável.

- Quando o vídeo é codificado a taxa de bits variável, diferentes quadros tem diferentes quantidades de dados. Um bloco de armazenamento pode conter diferentes números de quadro. Portanto, para implementar avanço e retrocesso rápidos, cada quadro deveria ser indexado individualmente.

- Quando o vídeo é codificado interquadro (explorando a redundância temporal dos quadros), como é o caso do MPEG, a decodificação de alguns quadros depende na disponibilidade de outros quadros. Uma opção para este caso é marcar quadros desejáveis para avanço e retrocesso rápidos e durante o avanço e retrocesso rápidos apenas estes quadros são usados.

QoS e interações com o usuário

Uma questão de QoS muito importante e pouco estudada é que quando o usuário altera o modo de apresentação do modo normal para outro modo (pausa, avanço e retrocesso rápidos), os requisitos de QoS para esta sessão alteram.

Durante a pausa, a sessão não é ativa por um tempo indeterminado. Se os recursos originalmente reservados para a sessão são chaveados para servir outra sessão, o usuário pode não ter recursos suficientes quando ele emite um comando de retomada e a apresentação não poderá ser retomada. Mas se os recursos são mantidos reservados durante a pausa, o usuário pode pagar por estes recursos mesmo se ele não está assistindo o filme.

Requisitos de QoS são normalmente aumentados durante operações de avanço e retrocesso rápidos, como visto anteriormente. Portanto, quando o usuário emite um comando de avanço ou retrocesso rápido, ele pode não ter recursos suficientes para executar a operação. Isto não é desejável, mas se cada sessão reservar recursos extras prevendo operações rápidas, estes recursos são desperdiçados na maior parte do tempo e o usuário paga mais por isto. Isto também não é interessante. A melhor solução é comprimir o vídeo e colocar os dados compactados de tal modo que nenhum requisito de QoS extra é necessário durante a operação rápida.