

Redes de Computadores
Trabalho Prático nº 2
(v0.9)

Sumário

Este trabalho consiste em desenvolver um cliente de um serviço de visualização de vídeo (filmes) inspirado nos princípios do MPEG-DASH (Dynamic Adaptable Streaming over HTTP). O filme está codificado num formato inspirado de Motion JPEG e está disponível via HTTP num servidor, codificado em várias resoluções. O cliente a desenvolver pelos estudantes deve visualizá-lo escolhendo dinamicamente a melhor resolução que seja adequada à capacidade efectiva da rede que o liga ao servidor, assim como à capacidade do próprio servidor HTTP.

1. Introdução

O objetivo deste trabalho é desenvolver um cliente de visualização de filmes (mudos), codificados num formato inspirado de Motion JPEG. Cada filme está acessível num servidor HTTP, codificado em diversas resoluções. O cliente deverá adaptar a resolução usada à capacidade efectiva da rede que o liga ao servidor, assim como à capacidade de serviço do servidor (número de pedidos por unidade de tempo que este consegue processar).

Cada filme está acessível num ficheiro com o nome do mesmo, seguido de um sufixo correspondente à resolução. Por exemplo, “o-dia-mais-curto-X.dat” em que X pode tomar, por exemplo, os valores 720p, 480p, 320p, 240p, 160p, correspondentes a diferentes resoluções. Estes ficheiros contêm uma sequência de registos (grosso modo equivalentes a fotogramas ou *frames* de um filme) com o seguinte formato:

Size em 32 bits, signed integer (int em Java)

TimeStamp em 64 bits, signed long (long em Java)

Imagem codificada em JPEG com *size* bytes (byte[] com *size* bytes em Java)

Cada registo contém uma imagem JPEG que ocupa um número variável de bytes, dependente da resolução e da compressão conseguida. Se os registos forem visualizados pelo cliente sequencialmente nos momentos indicados pela *timestamp*, o utilizador poderá visualizar corretamente o filme original. A *timestamp* é relativa à *timestamp* do primeiro registo e está expressa em nano segundos.

Dada a utilização de compressão, o filme constitui um *stream* do tipo VBR – *Variable Bit Rate* e requer uma capacidade variável da rede para ser transferido atempadamente. Esta capacidade está dependente das imagens e da resolução usada. Como é comum em clientes de *streaming*, é necessário dispor de um *buffer* local que acomode as variações da velocidade de transferência através da utilização de um *playout delay*.

Deve também ter em consideração que este formato particular é muito pouco eficiente, pois não comporta compressão temporal e a capacidade necessária para as diferentes resoluções é sempre muito significativa.

Como o filme está codificado em diferentes resoluções, existindo um ficheiro diferente para cada uma delas, o cliente pode tentar fazer o download dos registos correspondentes a cada pedaço de filme (um segmento temporal de alguns segundos) variando o ficheiro de que faz o download de cada segmento temporal, para assim ver se consegue evitar as situações em que precise de um registo para visualizar mas o mesmo ainda não chegou do servidor.

Assim, o seu cliente deve fazer o download do filme por segmentos, cada um correspondente a por exemplo 10, 20, 30, ... segundos de conteúdo, medir o tempo que leva a fazer o download dos segmentos, e verificar se pode aumentar a resolução (quanto maior melhor) ou se tem de a diminuir para a adaptar à capacidade da rede do servidor. Por prudência, e para poder usar um *playout delay* relativamente curto, no arranque o cliente pode começar por fazer o download de segmentos do filme na menor resolução. Abaixo dessa não é possível descer e se a mesma não permitir um download atempado dos segmentos, não será possível visualizar corretamente o filme, a não ser com um *playout delay* de muitos minutos.

Para fazer o download de um segmento (por exemplo do segundo 120 ao segundo 130) é necessário saber em que posição no ficheiro está o primeiro registo de cada segundo, ou seja, o seu *offset* no ficheiro. Para este efeito, associado a cada filme, com o mesmo nome do conteúdo e o sufixo “-index.dat” (como por exemplo “o-dia-mais-curto-index.dat”) existe um ficheiro que contém um conjunto de meta informação sobre o filme. Através deste ficheiro é possível conhecer o *offset*, no ficheiro correspondente a cada resolução, do início de cada segmento, seja qual for a sua dimensão temporal, desde que a mesma seja um múltiplo de 1 segundo. O formato exato do ficheiro *index* está descrito em anexo.

Ser-lhe-á fornecida uma classe de visualização que exporta um método que permite a visualização numa janela de uma imagem JPEG correspondente a um fotograma. Ser-lhe-á fornecida igualmente um exemplo da sua utilização. A invocação sucessiva deste método, respeitando as *timestamps*, com os vários fotogramas, permite visualizar corretamente o filme. Poderá também reutilizar o código desenvolvido ou fornecido nas aulas anteriores.

2. Requisitos e opções de funcionamento do cliente

A implementação do cliente deve obedecer aos requisitos a seguir descritos.

2.1 Especificação do cliente

O cliente deverá ser capaz de transferir e visualizar um filme usando a melhor resolução que for possível e será invocado da seguinte forma:

```
java MyDash URL-da-directoria-com-o-filme playout-delay segment-size
```

onde:

<URL-....> permite obter os ficheiros relativos ao filme em questão

<playout-delay> é o valor do playout delay em segundos

<segment-size> dimensão em segundos de cada segmento

2.3 Onde obter informação complementar

Java: <http://docs.oracle.com/javase/7/docs/api>

DASH: <http://en.wikipedia.org> (procurar MPEG-DASH)

2.4 Metodologia sugerida para o desenvolvimento do trabalho

Para a realização do trabalho deve seguir os passos inicialmente sugeridos na aula prática. A mesma passa pelas seguintes fases:

Fase 1. Inspirado na aula prática sobre *streaming* usando Multicast UDP, altere o código fornecido (Player+Viewer) e faça uma primeira versão do seu trabalho que mostre um filme numa resolução fixa a partir do disco, de modo a respeitar a temporização dos fotogramas. O nome do ficheiro com a resolução escolhida será o único parâmetro desta versão.

Fase 2. A partir do código desenvolvido na aula prática sobre HTTP, crie uma nova versão do programa anterior em que o conteúdo do ficheiro é obtido por HTTP, a partir do servidor fornecido. Esta versão terá como parâmetros o URL do ficheiro e o valor playout delay. Utilize pedidos HTTP na versão 1.0 (conexões não persistentes).

Fase 3. Adapte a versão anterior do seu programa de modo a separar a visualização dos fotogramas do seu carregamento. Para isso, pode usar o paradigma Produtor/Consumidor, onde um *thread* (Produtor) será responsável pelo download do conteúdo do ficheiro para uma fila de fotogramas. Um segundo *thread* (Consumidor) irá visualizar atempadamente os fotogramas contidos nessa fila.

Sugestão: Use a classe `java.util.concurrent.ConcurrentLinkedDeque` para implementar a fila de fotogramas à espera de serem reproduzidos. Antes de arrancar o Consumir poderá esperar que fila tem o correspondente ao playout delay desejado.

Fase 4. Expanda a versão anterior para fazer o download do ficheiro *index* e com base nessa informação fazer o download por segmentos do filme na resolução mais baixa, segmento a segmento.

Fase 5. Introduza a parte dinâmica e adaptativa do seu cliente.

Fase 6. Introduza as opções que achar mais interessantes (ou simples, ou possíveis, ...), algumas ilustradas na secção 4 a seguir.

3. Cenários de utilização

Avalie o funcionamento do seu trabalho executando o cliente e o servidor fornecido pelos docentes no mesmo computador e, eventualmente, noutros cenários indicados no modelo do relatório.

4. Funcionalidades mínimas e partes opcionais

Para que o seu trabalho seja aceite, o seu cliente deve ser capaz de executar as funcionalidades pedidas correspondentes pelo menos à fase 4 da proposta de desenvolvimento.

Opcionalmente, e de forma valorativa, o seu cliente pode ser capaz de:

- 1) Adaptar-se dinamicamente à capacidade da rede (fase 5)
- 2) Funcionar como proxy HTTP de Motion JPEG para um *player* (como por exemplo o VLC). Para realizar esta opção informe-se junto dos docentes sobre a especificação do formato Motion JPEG encapsulado em HTTP
- 3) O cliente usa HTTP 1.1 ao invés de HTTP 1.0
- 4) O cliente suporta que o utilizador faça *rewind* ou avance no filme
- 5) O cliente suporta mecanismos de segurança para o que deve pedir conselho aos docentes
- 6) Outras que ache interessantes e que deve discutir previamente com os docentes

5. Relatório do trabalho

Deverá elaborar um relatório (o modelo será fornecido pelos docentes) e no mesmo deverá indicar o que conseguiu implementar e os resultados que conseguiu obter, baseados no que o visualizador lhe mostrou pois apra além de visualizar os fotogramas este dá informações suplementares sobre a qualidade da reprodução.

ANEXO

Formato do ficheiro -index.dat

Abaixo encontra-se um extrato de um exemplo de ficheiro index.dat. O mesmo começa com várias linhas indicando quais as resoluções em que o filme está disponível. No exemplo existem 5 resoluções e portanto existirão 5 ficheiros terminados em *-160p.dat, *-240p.dat, *-320p.dat, *-480p.dat e *-720p.dat respectivamente, contendo outras tantas versões do filme em diferentes resoluções. Segue-se uma linha em branco que funciona como separador.

Em seguida aparece uma sequência de linhas com o formato geral:

resolução	número de segundos de filme	offset
-----------	-----------------------------	--------

Em cada linha, cada valor está separado do seguinte por espaço em branco. O número de segundos é relativo ao início do filme e o offset é a posição no ficheiro do 1.º byte correspondente ao registo (semelhante a um fotograma) onde começa o filme no segundo indicado. O offset é relativo ao início do ficheiro e o primeiro byte do mesmo tem o offset 0.

160p.dat		(comentário: esta é a primeira linha do ficheiro)
240p.dat		
320p.dat		
480p.dat		
720p.dat		
		(comentário: esta é uma linha em branco de separação)
160p.dat 1.0	64395	
240p.dat 1.0	105802	
320p.dat 1.0	153897	(comentário: o segundo 1.0 começa no offset 153897 no ficheiro *-480p.dat)
480p.dat 1.0	433926	
720p.dat 1.0	679626	
160p.dat 2.0	137229	
240p.dat 2.0	223118	
320p.dat 2.0	324854	
480p.dat 2.0	930734	
720p.dat 2.0	1453541	
.....		(comentário: foram suprimidas as linhas do segundo 3.0 ao segundo 263.0)
160p.dat 264.0	38574038	
240p.dat 264.0	67476001	
320p.dat 264.0	98201069	
480p.dat 264.0	320374221	
720p.dat 264.0	513140380	

Nota: Foi adicionado um último “registo” com índice 1.0e100 que corresponde ao comprimento do ficheiro...