

Redes de Computadores

Trabalho Prático 2

2015/2016

version 0.91

1 Introdução

O objetivo deste trabalho é desenvolver um cliente de reprodução de filmes inspirado nos princípios do MPEG-DASH(*) (Dynamic Adaptable Streaming over HTTP).

Um servidor HTTP disponibilizará filmes, codificados em diversas qualidades (*bitrates*). O cliente deverá adaptar, dinamicamente, a codificação usada na reprodução de um filme à capacidade efetiva 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).

Para a realização do trabalho são fornecidos:

- Um servidor HTTP (pygmy-httpd-trab2.jar) para alojar os filmes;
- Código fonte Java para reprodução do filme;
- Um filme codificado em várias qualidades, nos formatos adequados.

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

2 Especificações Gerais

Cada filme está acessível numa pasta, com o nome do filme, que conterá todos os ficheiros relativos ao mesmo.

Um filme consiste num ficheiro `index.dat` e um ou mais ficheiros multimédia, respeitantes a cada uma das qualidades de codificação ou bitrate disponibilizadas. Os ficheiros multimédia têm a extensão `.ts` e o seu nome é um número, correspondente à sua qualidade/*bitrate* (em kbps).

Por exemplo, o filme “*the-good-dinossaur*”, estará disponível em:

```
http://localhost:8080/the-good-dinossaur/  
                                index.dat  
                                128.ts  
                                256.ts  
                                512.ts
```

Os ficheiros multimédia estão codificados num formato não normalizado, pelo que não podem ser reproduzidos diretamente pelas ferramentas habituais. Será necessário desenvolver algum código antes de os poder entregar para reprodução pelo *player* fornecido.

2.1 Ficheiros Multimédia

Cada ficheiro multimédia `.ts` consiste numa sequência de segmentos de vídeo + áudio, de pequena duração, não necessariamente uniforme entre eles. Por exemplo, um filme com a duração de 120 segundos, terá aproximadamente 120 segmentos, se estes tiverem uma dimensão média de 1 segundo; mas, poderá haver segmentos mais longos ou mais curtos.

A sequência de segmentos é igual para as diferentes qualidades de reprodução, sendo os segmentos idênticos em tudo, exceptuando na sua dimensão (em *bytes*), decorrente da *bitrate* usada internamente para a codificação da *stream* de vídeo (o áudio não é afetado).

Como os segmentos das várias codificações são equivalentes, durante a reprodução do filme, um segmento de uma dada qualidade pode ser sucedido por um segmento de outra qualidade, superior ou inferior, sem que isso introduza quebras de continuidade quer no vídeo como no áudio. Porém, tal não dispensa a apresentação dos segmentos de forma atempada, para evitar pausas na reprodução.

Note-se que, neste trabalho, o processamento dos ficheiros `.ts` consistirá apenas em obter e decodificar os segmentos neles contidos. Para a decodificação do conteúdo multimédia contido nos segmentos (vídeo e áudio), será usado o código Java fornecido.

A especificação do formato dos ficheiros `.ts` é fornecida em anexo, no final deste documento.

2.2 Ficheiro Índice

O ficheiro `index.dat` tem dois propósitos. Em primeiro lugar, enumera as *bitrates* existentes para o filme. Ou seja, quais os ficheiros multimédia (e as respetivas qualidades) que poderão ser usados na sua reprodução. Em segundo lugar, enumera e descreve os segmentos que compõem o filme, especificamente: onde podem ser localizados dentro dos ficheiros multimédia `.ts` e a sua duração.

Com a informação contida no ficheiro `index.dat`, será possível solicitar ao servidor HTTP um ou vários segmentos (adjacentes) de um dado ficheiro multimédia `.ts`, através da realização de pedidos HTTP parciais, por conhecimento dos seus *byte offsets* dentro do ficheiro.

Não será, pois, necessário descarregar os ficheiros `.ts` na sua totalidade, podendo a reprodução do filme ser feita obtendo segmentos de qualquer dos ficheiros multimédia, em qualquer combinação que seja compatível com as condições da rede e carga do servidor.

A decisão informada, num dado momento da reprodução do filme, de qual dos ficheiros multimédia obter o(s) próximo(s) segmento(s) é uma questão central ao trabalho. Essa decisão terá que ser tomada com base na monitorização das condições da rede (desempenho dos pedidos HTTP já realizados) e o custo estimado de optar por um dos débitos disponíveis, sabendo a duração e comprimento dos segmentos, por consulta do ficheiro `index.dat`.

A especificação do formato dos ficheiros `index.dat` é fornecida em anexo, no final deste documento.

2.3 Solução Pretendida

O cliente a desenvolver deverá ser capaz de produzir um filme, procurando usar a melhor qualidade que for possível.

Deverá ser invocado da seguinte forma:

```
java    MyDash    <url> <playout-delay>

        <url> localização do filme no servidor
        <playout-delay> valor do playout delay em segundos
```

3 Metodologia de desenvolvimento

A realização do trabalho abrange as seguintes fases:

Fase 1. Adapte o código fornecido (`FilePlayer.java`) de modo a reproduzir um filme alojado no servidor HTTP (`pygmy-httpd-trab2.jar`).

O conteúdo multimédia de um dos ficheiros `.ts` será obtido de uma só vez. Utilize pedidos HTTP na versão 1.0 (conexões não persistentes). Esta versão terá como parâmetros o URL do ficheiro multimédia `.ts`. Ignore o valor do argumento *playout delay*.

Na formatação dos seus pedidos HTTP, nesta e nas próximas fases, deverá incluir o cabeçalho:

User-Agent: NNNNN-NNNNN

onde NNNNN será substituído pelos números de alunos dos elementos do grupo. Se o trabalho for realizado por apenas por um aluno, o número deverá ser repetido.

Fase 2. Expanda a versão anterior para fazer o download do filme, segmento a segmento, com base na informação (*offsets*) contida no ficheiro `index.dat`.

Esta versão terá como parâmetros o URL da pasta do filme e o *playout delay*. A reprodução do filme deverá esperar até terem sido obtidos segmentos suficientes para preencher o *playout delay*.

Fase 3. Adapte a versão anterior do seu programa de modo a separar a reprodução dos segmentos 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 segmentos. Um segundo *thread* (Consumidor) irá entregar atempadamente os segmentos contidos nessa fila para descodificação e reprodução pelo *media player* dado.

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

Fase 4. A produção do filme é feita obtendo segmentos aleatoriamente das duas qualidades mais baixas.

Fase 5. Introduza a parte dinâmica e adaptativa do seu cliente, seleccionando os segmentos e a qualidade de codificação com base nas condições observadas da rede.

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

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.

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

1. Adaptar-se dinamicamente à capacidade da rede (Fase 5)

2. O cliente usa HTTP 1.1 ao invés de HTTP 1.0
3. Funcionar como proxy HTTP para um *player* externo (como por exemplo o VLC). Para realizar esta opção informe-se junto dos docentes.
4. Outras que ache interessantes e que deve discutir previamente com os docentes

5 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.

6 Relatório do trabalho

Deverá elaborar um relatório (o modelo será fornecido pelos docentes e estará disponível no CLIP) e no mesmo deverá indicar o que conseguiu implementar e os resultados que conseguiu obter.

7 Entrega do trabalho

Este trabalho deverá ser entregue enviando uma mensagem de correio eletrónico para o endereço:
`rc.fct.unl.pt.2015@gmail.com`

com o seguinte *subject*: "Trabalho 2 dos alunos: n.º dos membros do grupo"

e tendo em anexo (pdf) com o relatório do trabalho (de acordo com a última versão do template que se encontra no sistema CLIP) e um arquivo (zip) com o código Java do trabalho do grupo até:
às 23h30 de 3 de Dezembro de 2015

O mini teste de avaliação do trabalho tem lugar depois do teste 2, na semana seguinte, e é obrigatório para todos os alunos que pretendam obter frequência este ano. O mesmo terá lugar na data e nas salas indicadas no CLIP.

Nota: o trabalho pode ser resolvido individualmente ou em grupo de no máximo 2 estudantes.

8 Observações

O código fornecido para execução do trabalho faz uso do pacote JavaFX8. Este pacote faz parte da distribuição Java 8 da Oracle/Sun, que difere da versão Java8 OpenJDK.

Certifique-se que tem instalada a versão mais recente do Java 8 da Oracle/Sun. À data de realização deste enunciado a versão mais recente consiste na release 66.

O Eclipse, por omissão, não permite a utilização imediata das classes do pacote JavaFX8. Este problema pode ser resolvido instalando o plugin EfxEclipse¹. Alternativamente, pode-se configurar o Eclipse, abrindo primeiro Preferences->Java->Compiler->Errors/Warnings e na opção `Deprecated and restricted API->Forbidden reference` seleccionar Ignore.

¹<http://www.eclipse.org/efxclipse/index.html>

9 Anexos - Formatos

9.1 Ficheiro Multimedia

O formato do ficheiro `.ts` é binário e consiste numa sequência de segmentos multimédia:

<segmento 1, segmento 2, segmento 3, ...>

Cada segmento consiste num registo com o seguinte formato binário:

<string, long, long, int, byte[]>

que significam, respetivamente:

- identificador do segmento, a ler com `readUTF()` de `DataInputStream`
- estampilha temporal, em mili-segundos, indicando quando reproduzir o segmento no *media player*;
- duração do segmento em mili-segundos;
- dimensão em bytes do vector seguinte;
- vector de bytes opaco, com a stream de vídeo e audio a entregar ao *media player*.

A sequência de segmentos do ficheiro multimédia termina com um segmento *dummy*, vazio, com identificador `eof`, que não deve ser entregue ao *media player*.

9.2 Ficheiro Índice

O ficheiro `index.dat` é um ficheiro de texto. As linhas começadas por `;` são comentários e devem ser ignoradas. O ficheiro tem duas partes, separadas por uma linha vazia. A parte inicial consiste na lista, ordenada, de *bitrates* existentes, da menor para a maior.

Depois da linha vazia, segue-se a enumeração dos segmentos, ordenada e repetida para cada *bitrate*.

Cada linha descreve um segmento. Os valores estão separados por um espaço. Indicam, respetivamente: o ficheiro `.ts` a que pertence o segmento, o número do segmento, o *byte offset* (deslocamento) para o primeiro *byte* do segmento no ficheiro `-ts`, e a sua duração em mili-segundos. Como se pode observar no exemplo abaixo.

```
; list of available bitrates, terminated by an empty line.
128.ts
256.ts
512.ts
1024.ts
2048.ts

; list of segments and respective offsets, sorted by encoding...
128.ts 0 0 1
128.ts 1 3820 1083
128.ts 2 59122 1000
...
linhas suprimidas
256.ts 0 0 1
256.ts 1 3820 1083
256.ts 2 87322 1000
...
linhas suprimidas
2048.ts 146 44534794 1000
2048.ts 147 44961775 500
2048.ts 148 44982488 0
```