Computação Avançada
Projeto de Avaliação

-

Mestrado em Engenharia Informática
Universidade do Minho
Relatório

Grupo	
PG41080	João Ribeiro Imperadeiro
PG41081	José Alberto Martins Boticas
PG41091	Nelson José Dias Teixeira

21 de Janeiro de 2020

Resumo

Este projeto de avaliação relativo à unidade curricular de Computação Avançada consiste, globalmente, em instalar e configurar um *cluster* de `HTCondor` com pelo 3 nós e utilizar o mesmo na resolução de uma tarefa de processamento (compressão de um vídeo, em *mp4*).

Conteúdo

Capítulo 1

Introdução

De forma sucinta, neste trabalho prático é pedida a implementação de um *cluster* `HTCondor` para a realização de tarefas de processamento de grandes volumes de dados ou de elevada complexidade. Para além disso, como seria de esperar, é necessário especificar o sistema desenvolvido pelos elementos deste grupo bem como aspetos adicionais associados ao mesmo.

Neste caso em concreto, é pedido algo mais específico, isto é, o desenvolvimento de uma aplicação de *resizing* de vídeo. Através deste caso, é possível mostrar o funcionamento do *cluster* em questão, transformando, por exemplo, um determinado vídeo com resolução *fullHD* (1080p) na resolução *SD* (720p).

A correta realização desta tarefa passa por reduzir o tempo necessário à conversão do vídeo em causa. Como tal, é possível partir o vídeo original em vários segmentos, fazendo a conversão de cada um destes e, no fim, juntá-los todos no vídeo de resultado. Seguindo a sugestão do docente desta unidade curricular, foi utilizado o comando *ffmpeg* para auxiliar a execução deste tipo de tarefa.

Capítulo 2

Implementação

2.1 Instalação e configuração do cluster

Antes de iniciarmos o desenvolvimento da solução para a realização da tarefa proposta, foi necessário configurarmos algumas máquinas, parte integrante do nosso *cluster*. Para isso, recorremos ao serviço *DigitalOcean*, onde é possível alugar máquinas virtuais. Assim, e dado que no enunciado são pedidas, no mínimo, 3 máquinas (valor este facilmente escalável a outros números de máquinas), alugamos 3 servidores virtuais (com 1 núcleo de processamento e $1\,GB$ de memória *ram*), os quais interligamos recorrendo a ligações provadas fornecidas pelo serviço.

Posto isto, fizemos a configuração inicial das máquinas, utilizando o sistema operativo `CentOS`, na versão 7, e seguimos os passos de instalação do `HTCondor`, ferramenta para criação de clusters, tal como indicado pelo docente num ficheiro na *Dropbox*. Tendo em conta que num sistema `HTCondor` há a noção de *master* e *worker*, tomámos a decisão de criar: 1 máquina *master*, que recebe os pedidos dos clientes e trata do processamento e eventual divisão de trabalho pelos *workers*, devolvendo uma resposta no fim do processamento; 3 máquinas *worker*, sendo que uma delas é a *master*, que estão responsáveis por efetuar tarefas pedidas pelo *master*.

Desta forma, o nosso cluster está configurado e pronto a receber tarefas para executar.

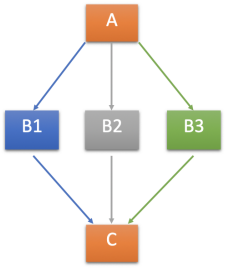
2.2 Realização da tarefa

Tendo o cluster configurado, o passo seguinte foi o desenvolvimento dos *scripts* que nos permitem executar a tarefa proposta. Esta tarefa consiste na receção de um vídeo na resolução 1080p de um utilizador pelo *master*. Por sua vez, o *master* procede à sua divisão em 3 partes iguais, enviando cada uma delas para cada um dos *workers*. Estes *workers* comprimem a sua parte do vídeo para a resolução 720p, tarefa esta que demora uma quantidade significativa de tempo, o que a torna ideal para este tipo de sistema de cluster. De seguida, os *workers* enviam a sua parte do vídeo, já comprimida, para o *master*, sendo que este junta todas as partes e devolve o vídeo comprimido ao cliente.

Para atingirmos este objetivo, proposto no enunciado, decidimos desenvolver vários *scripts*. O primeiro deles é um *script DAG*, que será submetido com o comando *condor_submit_dag* do `HTCondor`, e especifica a ordem das operações.

```
JOB A A.sub
JOB B1 B1.sub
JOB B2 B2.sub
JOB B3 B3.sub
JOB C C.sub
PARENT A CHILD B1 B2 B3
PARENT B1 B2 B3 CHILD C
```

(a) Código da tarefa



(b) Desenho da tarefa

Figura 2.1: Planificação da tarefa

No código acima, juntamente com o respetivo diagrama, podemos ver o nosso *script*, que indica ao `HTCondor` a ordem e a dependência das operações a realizar na tarefa. Em primeiro lugar, especificamos o trabalho *A*, onde o vídeo é divido em 3 partes iguais. De seguida, definimos 3 trabalhos, o *B1*, *B2* e *B3*, que correspondem ao processo de compressão de cada uma das 3 partes do vídeo. Como especificado no *script*, estes 3 trabalhos dependem do trabalho *A*, pois é necessário as 3 partes existirem antes de os *workers* as poderem comprimir. Por último, é definido o trabalho *C*, dependente do *B1*, *B2* e *B3*, que corresponde ao processo de junção das 3 partes comprimidas num só vídeo.

Passando agora às operações individuais, começamos por apresentar a de dividir o vídeo em 3 partes de igual comprimento.

```
// job A:

executable = ffmpeg
arguments = -i fragmento.mp4 -c copy -map 0 -segment_time 3.5 -f segment input%1d.mp4

should_transfer_files = YES
transfer_input_files = fragmento.mp4

when_to_transfer_output = ON_EXIT

queue 1
```

Como podemos ver, é utilizada uma das muitas vertentes da ferramenta `ffmpeg` para dividir o vídeo, sendo que o ficheiro do vídeo (fragmento.mp4) é indicado como sendo o *input*.

De seguida, temos as 3 operações de compressão das partes do vídeo. Dado que os 3 scripts são semelhantes, apresentaremos apenas um deles.

```
// job B1 (semelhante para os jobs B2 e B3):

executable = ffmpeg
arguments = -i input0.mp4 -s 1280x720 -c:a copy output0.mp4

should_transfer_files = YES
transfer_input_files = input0.mp4

when_to_transfer_output = ON_EXIT

queue 1
```

Mais uma vez, é utilizada a ferramenta `ffmpeg`, recebendo como *input* a parte correspondente a cada uma das tarefas, dependendo do *worker*.

Por fim, vem a operação de junção das 3 partes do vídeo. Como o `ffmpeg` é muito versátil, utilizamos mais uma das suas capacidades, permitindo-nos obter o vídeo pretendido.

```
// job C

executable = ffmpeg
arguments = -f concat -safe 0 -i outputs.txt -c copy compressed.mp4

should_transfer_files = YES
transfer_input_files = outputs.txt, output0.mp4, output1.mp4, output2.mp4

when_to_transfer_output = ON_EXIT

queue 1
```

Este *script* recebe como *inputs* um ficheiro contendo os caminhos para as diferentes partes do vídeo e as 3 partes do vídeo em si.

Capítulo 3

Conclusão

Para concluir, foi possível, de uma forma muito simples, configurarmos um cluster virtual utilizando o `HTCondor`, de forma a comprimir um vídeo, tarefa naturalmente muito exigente do ponto de vista computacional, ao dividirmos o trabalho por 3 máquinas diferente, acelerando este processo. Assim, foi-nos possível pormos em prática os conhecimentos adquiridos na parte de *HTC* da unidade curricular de Computação Avançada, o que nos permitiu experimentar com máquinas virtuais na *cloud* e as ferramentas `HTCloud`/`ffmpeg`. Por fim, consideramos que a nossa resolução foi muito positiva, ao utilizarmos diversas funcionalidades das ferramentas propostas e ainda termos utilizado o ambiente de máquinas virtuais do `HTCondor`.

Capítulo 4

Webgrafia

- Documentação - ***ffmpeg***:
  https://www.ffmpeg.org/ffmpeg.html
- Configuração do *cluster*:
  https://tinyurl.com/sr65bna