Gestão de Grandes Conjuntos de Dados

2º Trabalho Prático

-

Mestrado em Engenharia Informática
Universidade do Minho

Grupo nº 8	
PG41080	João Ribeiro Imperadeiro
PG41081	José Alberto Martins Boticas
PG41091	Nelson José Dias Teixeira
PG41851	Rui Miguel da Costa Meira

15 de maio de 2020

Conteúdo

Lista de Figuras

Capítulo 1

Introdução

Neste trabalho prático é requerida a concretização e avaliação experimental de tarefas de armazenamento e processamento de dados através do uso da ferramenta computacional *Spark* (*batch* e *streaming*). Por forma a realizar estas tarefas, são utilizados os dados públicos do *IMDb*, que se encontram disponíveis em:

https://www.imdb.com/interfaces/

Para além destes dados, é também utilizado um gerador de *streams*, baseado nos mesmos, que simula uma sequência de votos individuais de utilizadores. Este utensílio foi desenvolvido pelo docente desta unidade curricular e encontra-se disponível na plataforma *Blackboard*.

Ao longo deste documento vão também ser expostos todos os passos tomados durante a implementação das tarefas pedidas neste projeto, incluindo as decisões tomadas pelos elementos deste grupo a nível de algoritmos e parâmetros de configuração. Para além disso são ainda apresentadas todas as instruções que permitem executar e utilizar corretamente os programas desenvolvidos. Por fim, na fase final deste manuscrito, são exibidos os objetivos atingidos após a realização das tarefas propostas.

De salientar também que durante os capítulos que se seguem são identificadas algumas alternativas para concretizar as tarefas indicadas neste trabalho prático.

Capítulo 2

Implementação

Para a realização com sucesso deste trabalho prático, é solicitada a elaboração de três tarefas. Apresentam-se de seguida as mesmas:

1. Desenvolver uma componente de processamento de *streams* que produza os seguintes resultados:
   - ***Log***: armazenar todos os votos individuais recebidos, etiquetados com a hora de chegada aproximada ao minuto, em lotes de 10 minutos. Cada lote deve ser guardado num ficheiro cujo nome identifica o período de tempo;
   - ***Top3***: exibir a cada minuto o top 3 dos títulos que obtiveram melhor classificação média nos últimos 10 minutos;
   - ***Trending***: apresentar a cada 15 minutos os títulos em que o número de votos recolhido nesse período sejam superiores aos votos obtidos no período anterior, independentemente do valor dos votos.
2. Implementar uma componente de processamento em *batch* que permita realizar as seguintes tarefas:
   - ***Top10***: calcular o top 10 dos atores que participaram em mais títulos diferentes;
   - ***Friends***: computar o conjunto de colaboradores de cada ator (i.e., outros atores que participaram nos mesmos títulos);
   - ***Ratings***: atualizar o ficheiro *"title.ratings.tsv"* tendo em conta o seu conteúdo anterior e os novos votos recebidos até ao momento.
3. Escolher a configuração e a implementação que, para o mesmo *hardware*, permite receber e tratar o maior débito de eventos. Esta tomada de decisão deve ser devidamente justificada com recurso a resultados experimentais.

Nas próximas secções são evidenciadas as implementações para cada uma destas tarefas bem como algumas sugestões alternativas que poderiam ser tomadas em consideração.

2.1 Configuração

2.2 1ª Tarefa

2.2.1 *Log*

2.2.1.1 Alternativa

2.2.2 *Top3*

2.2.2.1 Alternativa

2.2.3 *Trending*

2.2.3.1 Alternativa

2.3 2ª Tarefa

2.3.1 *Top10*

Tal como foi mencionado no início 2º capítulo, nesta subtarefa é pedido o cálculo dos 10 atores que participaram em mais filmes distintos.

Durante o processamento inicial do ficheiro *"title.principals.tsv"* é, tal como seria de esperar, ignorado o respetivo cabeçalho. Posteriormente, é extraída, linha após linha, a informação pertinente do mesmo, isto é, os identificadores do filme e do ator em questão, agrupando os dados pela segunda componente. Esta última ação é efetuada com recurso à chamada do método *groupByKey*. Uma vez realizada esta computação, obtém-se para cada ator a lista de filmes em que este participou. Atendendo ao resultado exigido neste exercício, basta, nesta etapa do processamento, efetuar a contagem dos filmes associados a cada ator, filtrando os 10 registos com maiores valores.

A recolha dos 10 atores que participaram em mais filmes é formalizada com a chamada do método *top*. Esta função permite extrair os k maiores registos de um *RDD* segundo uma determinada ordem. Para o caso deste exercício, houve a necessidade de implementar um comparador explícito, numa classe à parte, dado que o tipo de dados *Tuple2* não é, por definição, serializável.

Tendo em consideração este último detalhe, conclui-se a realização desta subtarefa.

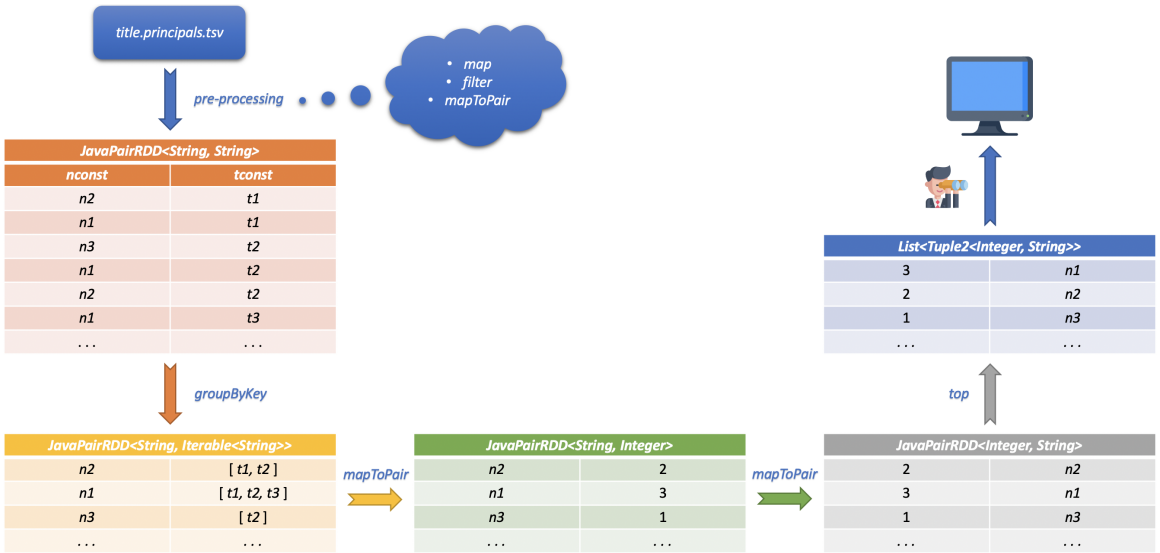

Figura 2.1: 2ª Tarefa (*batch*) - Esquema do processamento relativo à subtarefa *Top10*

2.3.1.1 Alternativa

Uma forma alternativa de resolver este exercício seria, na última fase do processamento, utilizar o método *take* em detrimento da função *top*. Esta escolha não foi tomada em consideração na implementação uma vez que o primeiro método necessita previamente que a informação esteja devidamente ordenada. Esta ordenação teria de ser realizada com a invocação do método *sortByKey(false)*, colocando a contagem dos filmes em que cada ator participou de forma decrescente. Este último facto representa uma ineficiência no cálculo do resultado pretendido uma vez que é efetuada a ordenação completa da informação em causa e, para além disso, realiza-se desnecessariamente um passo computacional extra.

2.3.2 *Friends*

Neste exercício é requerido a computação do conjunto de colaboradores associado a cada ator, ou seja, o grupo dos atores que particapam nos mesmos filmes.

Durante o processamento inicial do ficheiro *"title.principals.tsv"* é, tal como seria de esperar, ignorado o respetivo cabeçalho. Posteriormente, é extraída, linha após linha, a informação pertinente do mesmo, isto é, os identificadores do filme e do ator em questão, agrupando os dados pela primeira componente. Esta última ação é efetuada com recurso à chamada do método *groupByKey*. De forma a obter o resultado solicitado nesta subtarefa, é necessário, nesta fase da computação, proceder à realização de uma operação denominada por produto cartesiano. Nesta operação computa-se, num dado momento, vários pares de atores que coloboraram num determinado filme. Uma vez realizado este cálculo, é invocado novamente o método *groupByKey* de forma a obter o resultado pretendido, isto é, o conjunto de colaboradores para cada ator presente nos dados públicos do *IMDb*.

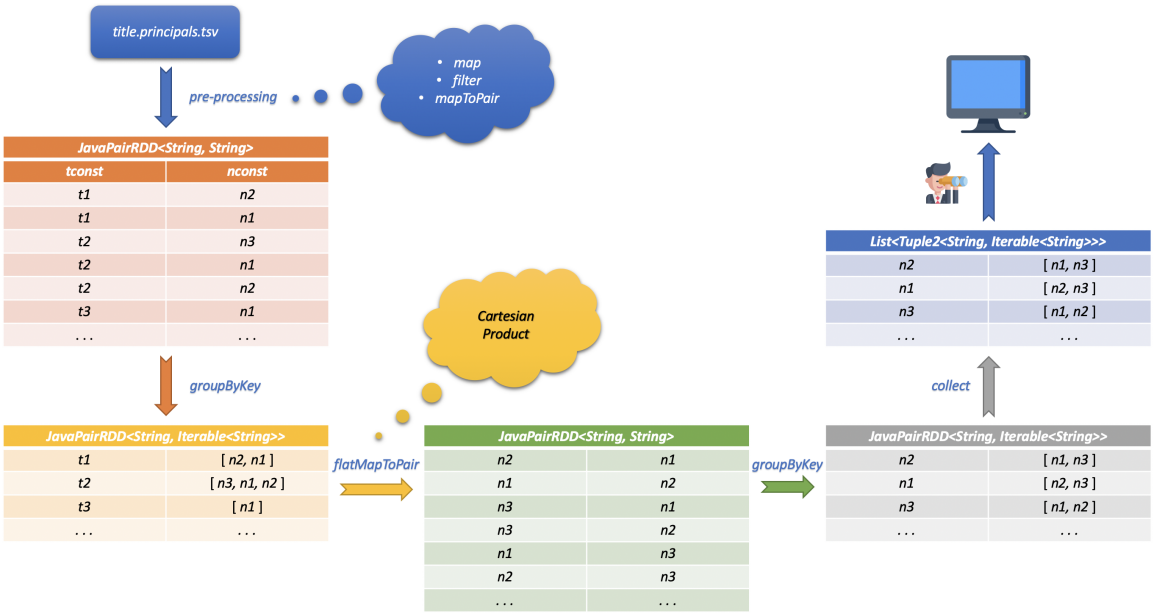

Figura 2.2: 2ª Tarefa (*batch*) - Esquema do processamento relativo à subtarefa *Friends*

2.3.2.1 Alternativa

2.3.3 *Ratings*

2.3.3.1 Alternativa

2.4 3ª Tarefa

Capítulo 3

Conclusão

Apêndice A

Observações

- Documentação *Java* 8:
  https://docs.oracle.com/javase/8/docs/api/
- *Maven*:
  https://maven.apache.org/
- *Apache Spark*:
  https://spark.apache.org/
- *Docker*:
  https://www.docker.com/