MC970/MO644 - Programação Paralela
Laboratório 11

Professor: Guido Araújo
Monitor: Hervé Yviquel

Analisador de Texto

Neste laboratório, iremos implementar um analisador/comparador de textos usando Apache Spark.

Enunciado

Neste exercício o objetivo é desenvolver uma analisador de texto usando Spark baseado na contagem de palavras que foi apresentada na aula. Para facilitar o exercício, o esqueleto do projeto é fornecido. O programa tem como argumento o caminho completo para dois arquivos de texto para analisar. O programa é divido em 2 partes:

- A primeira parte do programa deve contar as palavras e imprimir na saída padrão as 5 palavras de mais de 3 letras que tem as maiores ocorrências em ordem decrescente para cada texto. O programa deve considerar letras maiúsculas e minúsculas como sendo a mesma letra, além disso deve ignorar caracteres de pontuação para contar as ocorrências (usando replaceAll ("[,.!?:;]","")). Aconselhamos de começar para dividir o texto a partir dos espaços, depois *limpar* a pontuação, e finalmente contar as palavras. Nota-se que o regex fornecido para limpar as palavras da pontuação não considera todos os casos mas produze as saídas esperadas por Parsusy.

- Na segunda parte, o programa deve comparar as analises dos 2 textos, e imprimir em ordem alfabética todas as palavras que aparecem mais de 100 vezes em cada um dos dois textos.

Caso tenha alguma dúvida, use o Google Groups - para este trabalho está liberado discutir a solução direta do problema.

Testes e Resultado

Para compilar o seu programa, pode usar qualquer computador com Spark e sbt instalado (incluindo o servidor mo644 e as máquinas dos labs do IC), e digitar o comando seguinte na pasta própria do programa (contendo o arquivo *build.sbt*):

```
$ sbt package
```

Para executá-lo, basta digitar no mesmo computador:

```
$ spark-submit --class Analisador \
        target/scala-2.11/analisador_2.11-0.1.jar \
        texto1.txt texto2.txt
```

Os testes serão executados em 3 textos abertos e 1 fechado. Todas as combinações de textos serão testadas (então faz 3 testes abertos e 3 fechados). O programa deve imprimir os resultados na saída padrão respeitando o formato. As saídas esperadas dos testes abertos são fornecidas. Os arquivos de entrada contem somente texto sem formatação .

Submissões

O número máximo de submissões é de 10. A submissão deve ser o arquivo Scala.

Compilação e Execução

O ParSuSy irá compilar o seu programa usando o compilador de Scala.