

6º ATIVIDADE de CES-27 / 2018

CTA - ITA - IEC

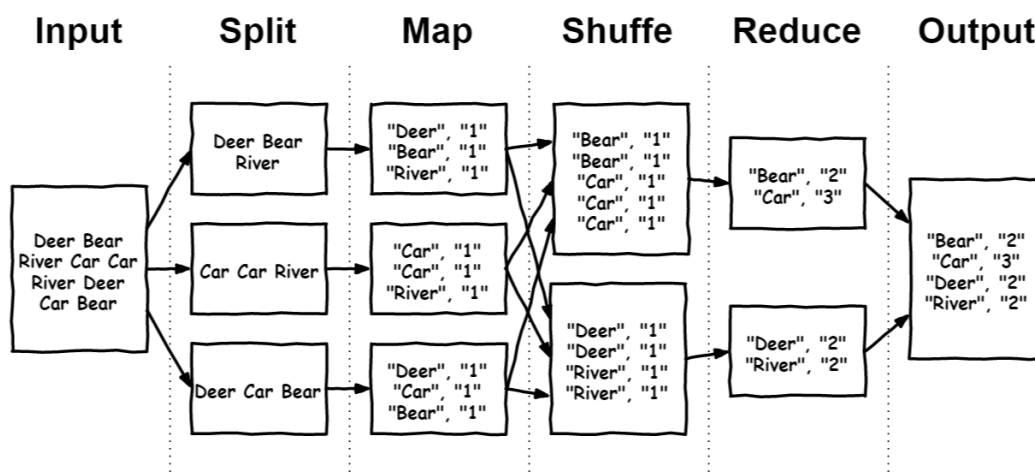
Prof Juliana e Prof Vitor

Importante: É permitido fazer essa atividade individualmente ou em dupla.

Objetivo: Trabalhar com o modelo de programação MapReduce de forma sequencial.

Entregar (através do TIDIA): Códigos dos exercícios (arquivos .go) e relatório. O relatório deve apresentar o código, explicar detalhes particulares/críticos do código, apresentar testes realizados e comentar resultados.

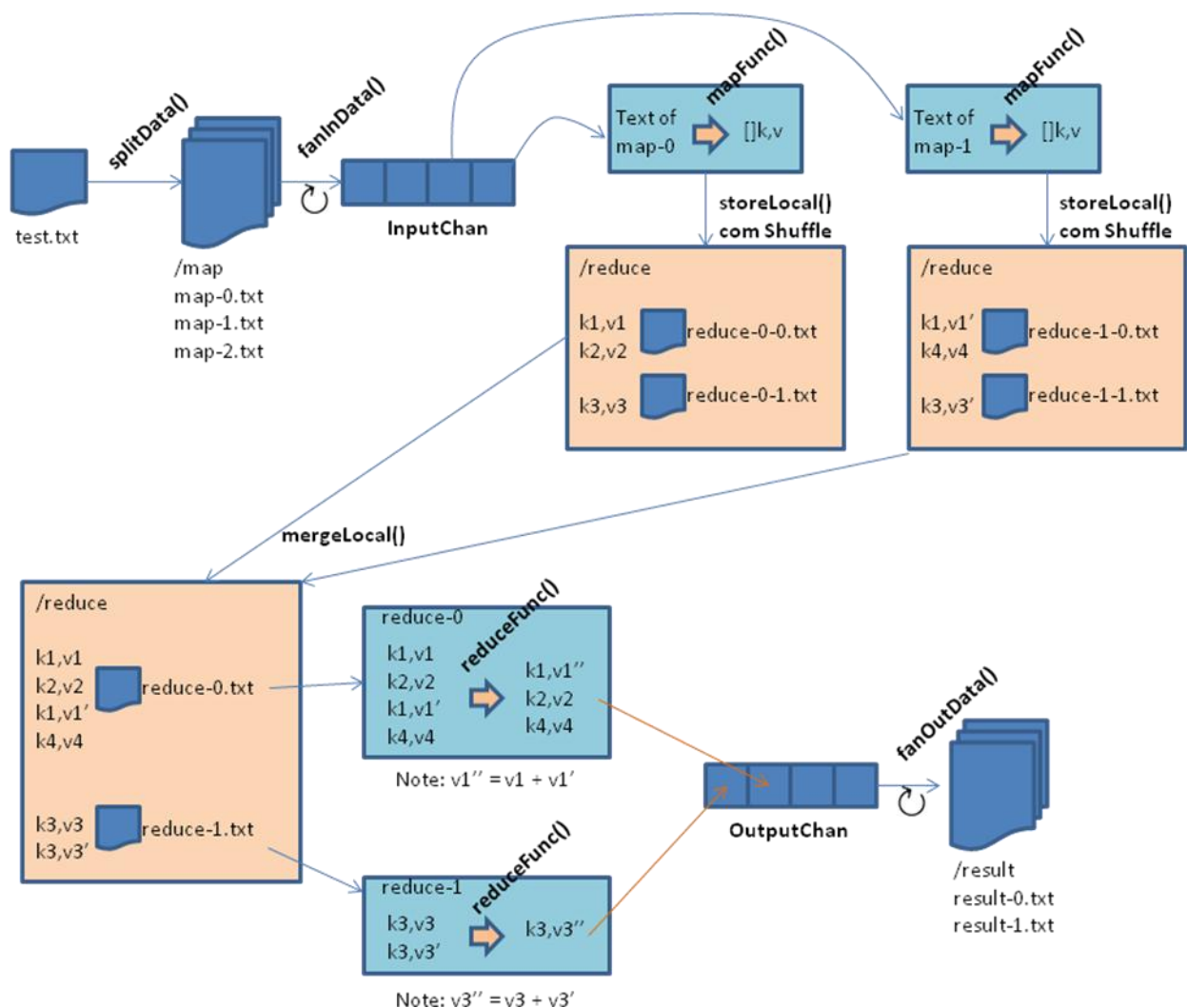
Processo de MapReduce



Primeiramente vamos compreender o framework fornecido.

- Baixar os códigos labMapReduce.zip
- Descompactar essa pasta no seu \$GOPATH/src
 - Deve ficar a pasta labMapReduce com duas outras dentro (mapreduce e wordcount)
- Para instalar package mapreduce fornecida:
 - Entrar em \$GOPATH/src/labMapReduce/mapreduce:
 - Executar: go install
 - Verificar que aparece mapreduce.a no subdiretório /pkg de \$GOPATH
 - Esse passo é necessário para os arquivos do package wordcount poder importar "labMapReduce/mapreduce" corretamente.
- Para compilar o package wordcount:
 - Entrar em \$GOPATH/src/labMapReduce/wordcount
 - Executar: go build
- Para rodar o programa:
 - Entrar em \$GOPATH/src/labMapReduce/wordcount
 - Executar: wordcount.exe -file files/test.txt
 - Obs: Esse arquivo test.txt é o que será processado
- Na função main.go, está definido 2 reducejobs (ou seja, simularemos 2 reduce workers) e chunksize de 1024 bytes (ou seja, esse é o tamanho máximo de cada “pedaço” do arquivo original que será “fatiado”).

Esquema para ajudar a compreender o funcionamento do programa:



Questão 1: Implemente a função mapFunc para identificar as palavras de cada fatia do arquivo inicial. Você pode trabalhar de suas formas:

- Registrar toda palavra com valor um. Ex: <"Car",1> <"River", 1> <"Car",1>
- Ou já registrar a soma parcial. Ex: <"Car", 2> <"River", 1>

Obs: Não precisa ter ordenação entre as palavras. Ou seja, podia ser <"River", 1> <"Car", 2>

Questão 2: Implemente a função reduceFunc para consolidar a contagem das palavras.

Obs: Não precisa ter ordenação entre as palavras.

Questão 3: Rode o programa com o arquivo test.txt e comente sobre os resultados (parciais e finais) indicando se estão coerentes com o esperado.

Questão 4: Rode o programa com outro arquivo. Se desejar, você pode alterar o valor de reducejobs e chunksize. Comente os resultados.

Bom trabalho!