Tesseract

Envio local, exemplos de scripts

Informações sobre o uso direto do cluster local estão em [[SLURM|aqui]].

Exemplo simples

Exemplo para processar todos os .tiff de um diretório usando apenas um core (serial). Verifica se já existe um arquivo processado antes.

ocr.sh:

```
#!/bin/bash
module load tesseract

for INPUT in *.tiff
do
  BASENAME=$(basename $INPUT .tiff)

  if [ ! -f "${BASENAME}.pdf" ]
  then
    echo "Processing $INPUT"
    tesseract "$INPUT" "$BASENAME" -l por pdf
  fi
done



sbatch ocr.sh
```

Exemplo paralelo

Exemplo para processar todos os .tiff de um diretório usando múltiplos cores (paralelo).

A ideia desse script é permitir que vários processos independentes trabalhem no mesmo diretório sem definir previamente qual processo ficará responsável por cada arquivo. O workflow é:

1. O processo é iniciado e lista todos os arquivos com uma extensão (.tiff, no exemplo)
2. Para cada arquivo encontrado
    1. Verifica se ele já foi convertido (verificando se existe um .pdf)
    2. Verifica se existe um arquivo de lock (ou seja, algum outro processo está trabalhando nesse arquivo)
    3. Se não houver:
        1. cria o arquivo lock com o seu identificador e verifica depois de 5 segundos se o lock ainda é seu. (Isso é feito para evitar problemas de concorrência.)
        2. Converte o arquivo gerando um arquivo temporário
        3. Renomeia o arquivo temporário para o nome final.
        4. Remove o lock

```
#!/bin/bash
module load tesseract

for INPUT in *.tiff
do
  BASENAME=$(basename $INPUT .tiff)

  if [ ! -f "${BASENAME}.pdf" ] && [ ! -f "${BASENAME}.lock" ]
  then
    # lock and wait 5 seconds (to reduce concurrency)
    echo "$SLURM_ARRAY_TASK_ID" > "${BASENAME}.lock"
    sleep 5
```

```
   if [ "$(cat ${BASENAME}.lock)" = "$SLURM_ARRAY_TASK_ID" ]
   then
    echo "Processing $INPUT -- (task $SLURM_ARRAY_TASK_ID)"
    TMPNAME=".$BASENAME-$SLURM_ARRAY_TASK_ID"

    tesseract "$INPUT" "$TMPNAME" -l por pdf

    mv "$TMPNAME.pdf" "$BASENAME.pdf"
    rm "${BASENAME}.lock"
   else
    echo "Lost lock on $INPUT"
   fi
  fi
done
```

Para usar 30 cores:

```
sbatch --array=0-30 ocr.sh
```

Ghostscript (.pdf -> .tiff)

Exemplo para processar todos os .pdf de um diretório usando múltiplos cores (paralelo).

A ideia desse script é permitir que vários processos independentes trabalhem no mesmo diretório sem definir previamente qual processo ficará responsável por cada arquivo. O workflow é:

```
#!/bin/bash

for INPUT in *.pdf
do
  BASENAME=$(basename $INPUT .pdf)

  if [ ! -f "${BASENAME}-001.tiff" ] && [ ! -f "${BASENAME}.lock" ]
  then
    # lock and wait 5 seconds (to reduce concurrency)
    echo "$SLURM_ARRAY_TASK_ID" > "${BASENAME}.lock"
    sleep 5

    if [ "$(cat ${BASENAME}.lock)" = "$SLURM_ARRAY_TASK_ID" ]
    then
      echo "Processing $INPUT -- (task $SLURM_ARRAY_TASK_ID)"

      gs -q -dNOPAUSE -dBATCH -sDEVICE=tiffg4 -r400x400 -sOutputFile=${BASENAME}-%03d.tiff ${INPUT}

      rm "${BASENAME}.lock"
    else
      echo "Lost lock on $INPUT"
    fi
  fi
done
```

Para usar 30 cores:

```
sbatch --array=0-30 gs.sh
```