In [2]:

```
#                          import libraries, view the content of the externally (
locally) mounted directory
import os
import sys
import numpy as np
import pandas as pd

cromwell_dir = '/usr/local/etc'
if os.path.isdir(cromwell_dir) == False:
    print('Cromwell not found:')
else:
    print('Womtool and Cromwell jar files')
    print(os.listdir(cromwell_dir))

external_mount = os.getcwd()
os.listdir(external_mount)
```

```
Womtool and Cromwell jar files
['cromwell-36.jar', 'womtool-36.jar', 'jupyter']
```

Out[2]:

```
['Jsons',
 '.DS_Store',
 'Config',
 'Dockerfile',
 'Makefile',
 'wiki_example notebook.ipynb',
 'test.txt',
 '.ipynb_checkpoints']
```

## Write files in this "perishable directory"

Or upload this notebook to the ../tmp_usr/run_dir and output will be saved in the mounted drive.

In [3]:

```
%%writefile AAA_json.json
{
    "wf_hello.ImpString": [
                           [ ["0:0:0", "0:0:1"],
                             ["0:1:0","0:1:1"] ],
                           [ ["1:0:0", "1:0:1"],
                             ["1:1:0","1:1:1"] ],
                           [ ["2:0:0", "2:0:1"],
                             ["2:1:0","2:1:1"] ] ]
}
```

```
Writing AAA_json.json
```

In [5]:

```
%%writefile AAA_T2IdxArray.wdl
###########################
#              AAA_T2IdxArray.wdl
# Usage:
# java -jar full/path/to/cromwell run AAA_T2IdxArray.wdl -i AAA_json.json
# wf_hello.ImpString = [[["1"]]]
###########################

task hello {
  String addressee
  Int One
  Int Two
  Int Three
  Int sIx
  command {
    echo "Input String =  ${addressee} With Iterator Index: ${sIx}  [${One}][${Two}][${Three}]"
  }
  output {
    String message = read_string(stdout())
  }
}

workflow wf_hello {

  Array[Array[Array[String]]] ImpString
  Int arr1_size = length(ImpString)
  Int arr2_size = length(ImpString[0])
  Int arr3_size = length(ImpString[0][0])

  Array[Int] Index = range(arr1_size * arr2_size * arr3_size)

  scatter (idx in Index) {

    Int reads_idx = idx % arr3_size

    Int lanes_idx = (idx / arr3_size) % arr2_size
    Int lanes_idx = (idx / arr2_size) % arr2_size

    Int samples_idx = (idx / (arr3_size * arr2_size)) % arr1_size

    call hello as howdy {
      input:
        addressee = ImpString[samples_idx][lanes_idx][reads_idx],
        One = samples_idx,
        Two = lanes_idx,
        Three = reads_idx,
        sIx = idx
    }
  }

  output {
     howdy.message
  }
}
```

```
Overwriting AAA_T2IdxArray.wdl
```

In [6]:

```
run_string = 'java -jar /usr/local/etc/cromwell-36.jar run AAA_T2IdxArray.wdl -i
AAA_json.json'
os.system(run_string)
```

Out[6]:

```
0
```

In [7]:

```
os.listdir('cromwell-executions/wf_hello')
```

Out[7]:

```
['ad36b6b5-8bdc-46a9-b4d4-26bba515a727']
```

In [8]:

```
ex_path = 'cromwell-executions/wf_hello/ad36b6b5-8bdc-46a9-b4d4-26bba515a727/call-howdy'
os.listdir(os.path.join(ex_path, 'shard-0/execution'))
```

Out[8]:

```
['stdout',
 'script.background',
 'stderr',
 'script.submit',
 'script',
 'stdout.background',
 'rc',
 'stderr.background']
```

In [9]:

```
shards = [os.path.join(ex_path, d + '/execution') for d in os.listdir(ex_path)]
print('%i shards found\nwith stdout files:\n\n'%(len(shards)))
for shard in shards:
    with open(os.path.join(shard, 'stdout'), 'r') as fh:
        lines = fh.readlines()
    for line in lines:
        print(line.strip())
```

```
12 shards found
with stdout files:


Input String =  2:0:1 With Iterator Index: 9  [2][0][1]
Input String =  1:1:1 With Iterator Index: 7  [1][1][1]
Input String =  0:0:0 With Iterator Index: 0  [0][0][0]
Input String =  0:0:1 With Iterator Index: 1  [0][0][1]
Input String =  1:1:0 With Iterator Index: 6  [1][1][0]
Input String =  2:0:0 With Iterator Index: 8  [2][0][0]
Input String =  2:1:1 With Iterator Index: 11  [2][1][1]
Input String =  2:1:0 With Iterator Index: 10  [2][1][0]
Input String =  0:1:1 With Iterator Index: 3  [0][1][1]
Input String =  1:0:0 With Iterator Index: 4  [1][0][0]
Input String =  1:0:1 With Iterator Index: 5  [1][0][1]
Input String =  0:1:0 With Iterator Index: 2  [0][1][0]
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: