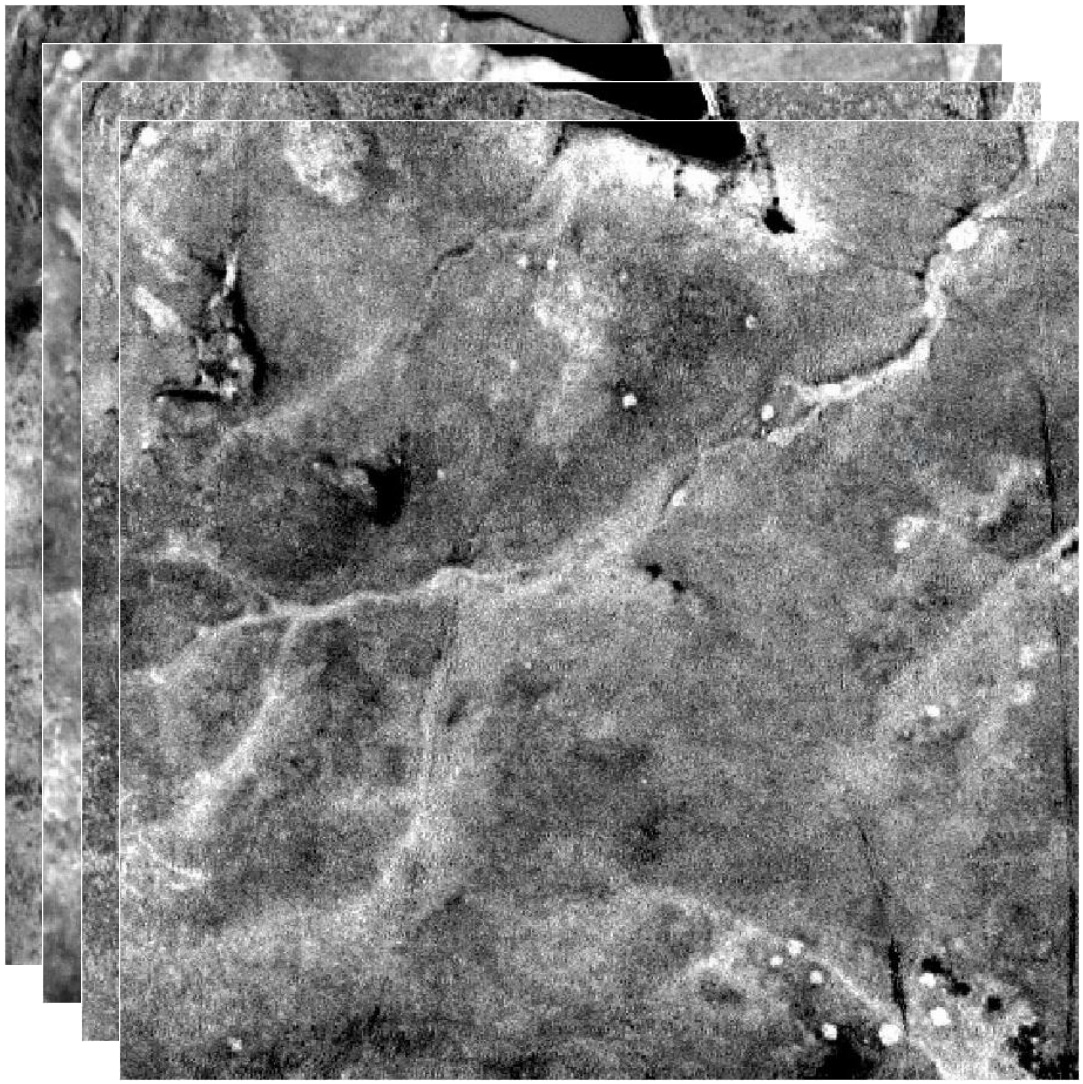
Parallelizing a Python Geoprocessing Tool

GeoDev Meetup - Seattle, WA
April 8th, 2015

David Howes, Ph.D. - David Howes, LLC
dhowes.com

Eric Sant - Open Range Consulting
openrangeconsulting.com

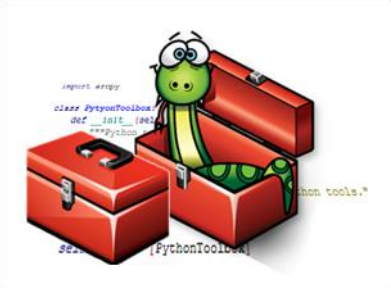
Task: Run Dependent Con Statements Using Conditions Data from the Statistical Package R

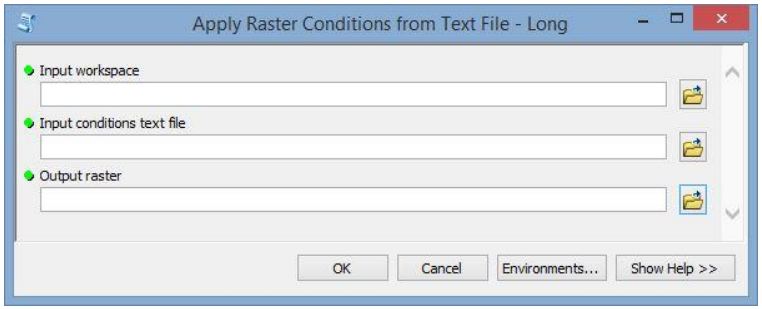


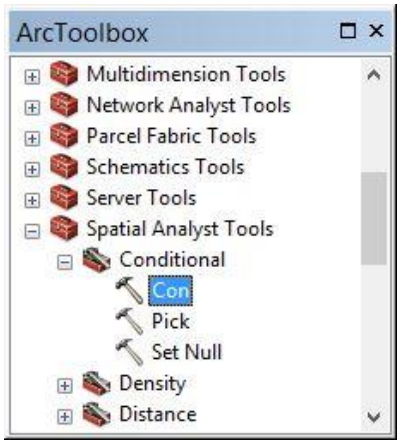
Use ArcGIS Geoprocessing Tool to Create & Run Con Statements

sh_model_short.txt

```
1) root 281 21.3600 0.32300
  2) BGW3_MEAN < 92.6113 41  1.3770 0.86840 *
  3) BGW3_MEAN > 92.6113 240  5.7040 0.22980
    6) BGW3_MEAN < 105.116 57  1.2090 0.39220 *
    7) BGW3_MEAN > 105.116 183  2.5230 0.17920
     14) BGW4_MEAN < 158.219 132  1.5270 0.21660 *
     15) BGW4_MEAN > 158.219 51  0.3352 0.08251 *
```







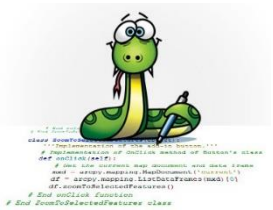

```
temp0 = Con((Raster("BGW3") > 92.6113) & (Raster("BGW3") > 105.116) & (Raster("BGW4") > 158.219), 0.08251)
temp1 = Con((Raster("BGW3") > 92.6113) & (Raster("BGW3") > 105.116) & (Raster("BGW4") < 158.219), 0.21660, temp0)
temp2 = Con((Raster("BGW3") > 92.6113) & (Raster("BGW3") < 105.116), 0.39220, temp1)
temp3 = Con((Raster("BGW3") < 92.6113), 0.86840, temp2)
temp3.save("C:\\Temp\\ARC_Out_Part_1")
```

Problem: Tool is slow for big images with thousands of Con statements

So here's how to make the process 50% faster (with a caveat)...

Run Apply Raster Conditions Tool Outside ArcMap

run_arc_tool.py



Read input file

ARC_Input_Full_Small.txt

```
toolboxPath|C:\Temp\ORC_Tools_1_3.pyt
inWorkspacePath|C:\Temp\Full_Small.gdb
conditionsFilePath|C:\Temp\sh_model_short.txt
outWorkspacePath|C:\Temp\Full_Small_Out.gdb
outRasterName|ARC_Out
tempFolderPath|C:\Temp
messagesFilePath|C:\Temp\ARC_Messages.txt
extractRasters|False
appendRasters|True
showMessages|False
```

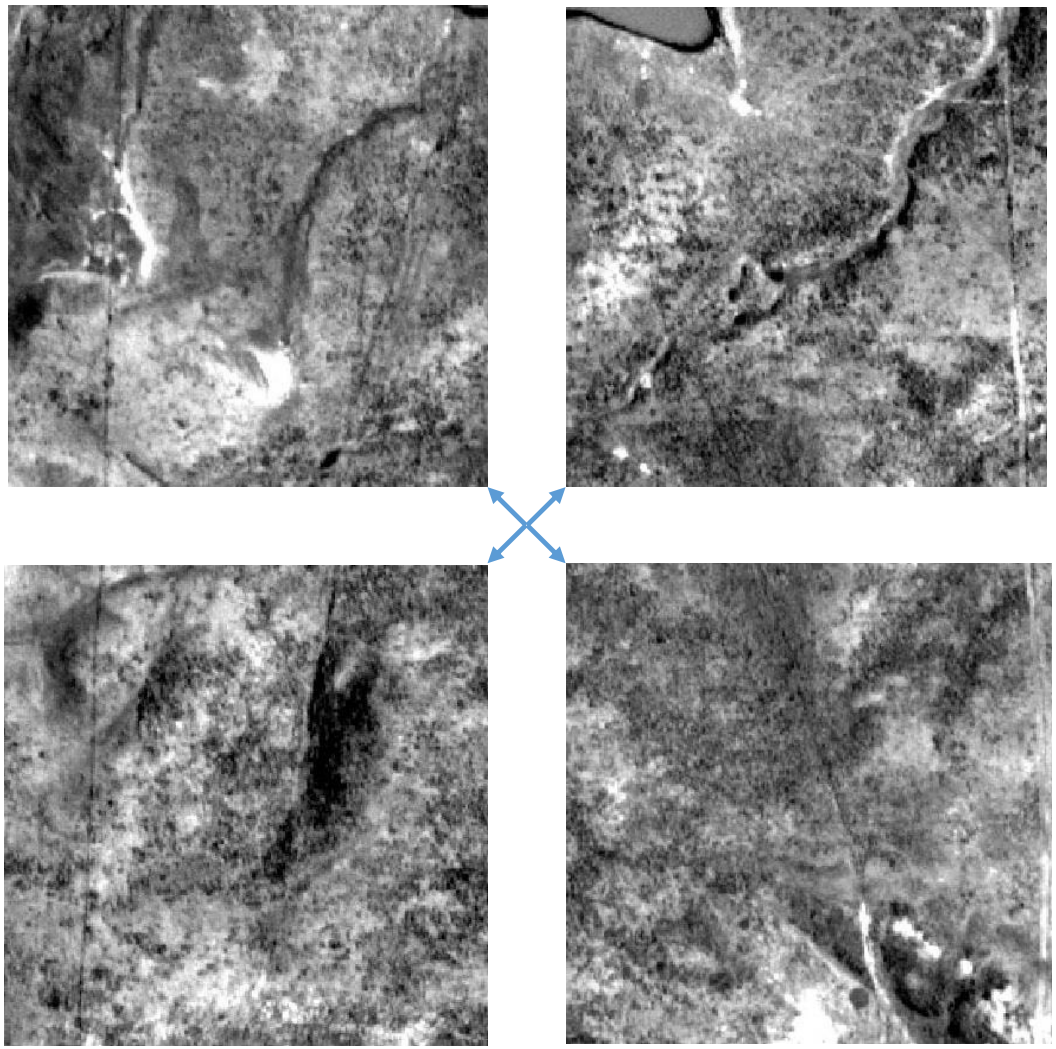
```
# Import toolbox
arcpy.ImportToolbox(toolboxPath)

# Run tool
arcpy.ApplyRasterConditionsTool_ORCTools(inWorkspacePath, conditionsFilePath, outRasterPath)

# Store geoprocessing messages
```

Store Geoprocessing Messages

ARC_Messages.txt

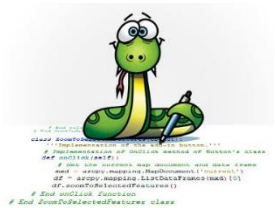
```
Executing: ApplyRasterConditionsTool "C:\Temp\Part_In_1.gdb" "C:\Temp\sh_model_short.txt" "C:\Temp\Full_Small.gdb\ARC_Out"
Start Time: Fri Apr 03 16:55:55 2015
Running script ApplyRasterConditionsTool...
PARAMETERS:
workspace_path: C:\Temp\Full_Small.gdb
in_conditions_file: C:\Temp\sh_model_short.txt
out_raster_path: C:\Temp\Full_Small_Out.gdb\ARC_Out
Checking input data...
Preparing input data...
Building Con expressions...
(Raster("BGW4") > 158.219)
(Raster("BGW3") > 105.116)
(Raster("BGW3") > 105.116) & (Raster("BGW4") > 158.219)
(Raster("BGW3") > 92.6113)
(Raster("BGW3") > 92.6113) & (Raster("BGW3") > 105.116) & (Raster("BGW4") > 158.219)
Con((Raster("BGW3") > 92.6113) & (Raster("BGW3") > 105.116) & (Raster("BGW4") > 158.219), 0.08251
(Raster("BGW4") < 158.219)
(Raster("BGW3") > 105.116)
(Raster("BGW3") > 105.116) & (Raster("BGW4") < 158.219)
(Raster("BGW3") > 92.6113)
(Raster("BGW3") > 92.6113) & (Raster("BGW3") > 105.116) & (Raster("BGW4") < 158.219)
Con((Raster("BGW3") > 92.6113) & (Raster("BGW3") > 105.116) & (Raster("BGW4") < 158.219), 0.21660
(Raster("BGW3") < 105.116)
(Raster("BGW3") > 92.6113)
(Raster("BGW3") > 92.6113) & (Raster("BGW3") < 105.116)
Con((Raster("BGW3") > 92.6113) & (Raster("BGW3") < 105.116), 0.39220
(Raster("BGW3") < 92.6113)
Con((Raster("BGW3") < 92.6113), 0.86840
Running Con expressions...
temp0 = Con((Raster("BGW3") > 92.6113) & (Raster("BGW3") > 105.116) & (Raster("BGW4") > 158.219), 0.08251)
temp1 = Con((Raster("BGW3") > 92.6113) & (Raster("BGW3") > 105.116) & (Raster("BGW4") < 158.219), 0.21660, temp0)
del temp0
temp2 = Con((Raster("BGW3") > 92.6113) & (Raster("BGW3") < 105.116), 0.39220, temp1)
del temp1
temp3 = Con((Raster("BGW3") < 92.6113), 0.86840, temp2)
del temp2
temp3.save("C:\\Temp\\Full_Small_Out.gdb\\ARC_Out")
del temp3
Running Con expressions...
Completed script ApplyRasterConditionsTool...
Succeeded at Fri Apr 03 16:55:59 2015 (Elapsed Time: 6 minutes 22 seconds)
```

Split Input Rasters into Parts and Process Simultaneously



Run Apply Raster Conditions Tool in Parallel

run_arc_parallel.py



Read input file

ARC_Input_Full_Small.txt

```
toolboxPath|C:\Temp\ORC_Tools_1_3.pyt
inWorkspacePath|C:\Temp\Full_Small.gdb
conditionsFilePath|C:\Temp\sh_model_short.txt
outWorkspacePath|C:\Temp\Full_Small_Out.gdb
outRasterName|ARC_Out
tempFolderPath|C:\Temp
messagesFilePath|C:\Temp\ARC_Messages.txt
extractRasters|False
appendRasters|True
showMessages|False
```

Split input rasters into parts

For each part

Create input file

ARC_Input_Full_Small_Part_1.txt

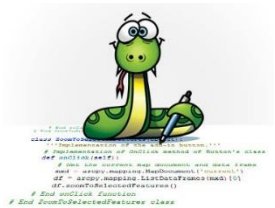
```
toolboxPath|C:\Temp\ORC_Tools_1_3.pyt
inWorkspacePath|C:\Temp\Part_In_1.gdb
conditionsFilePath|C:\Temp\sh_model_short.txt
outWorkspacePath|C:\Temp\Part_Out_1.gdb
outRasterName|ARC_Out_Part_1
messagesFilePath|C:\Temp\ARC_Messages_Part_1.txt
```

Call **process_arc_part.py** - sets up and runs **run_arc_tool.py** in parallel

Append output rasters

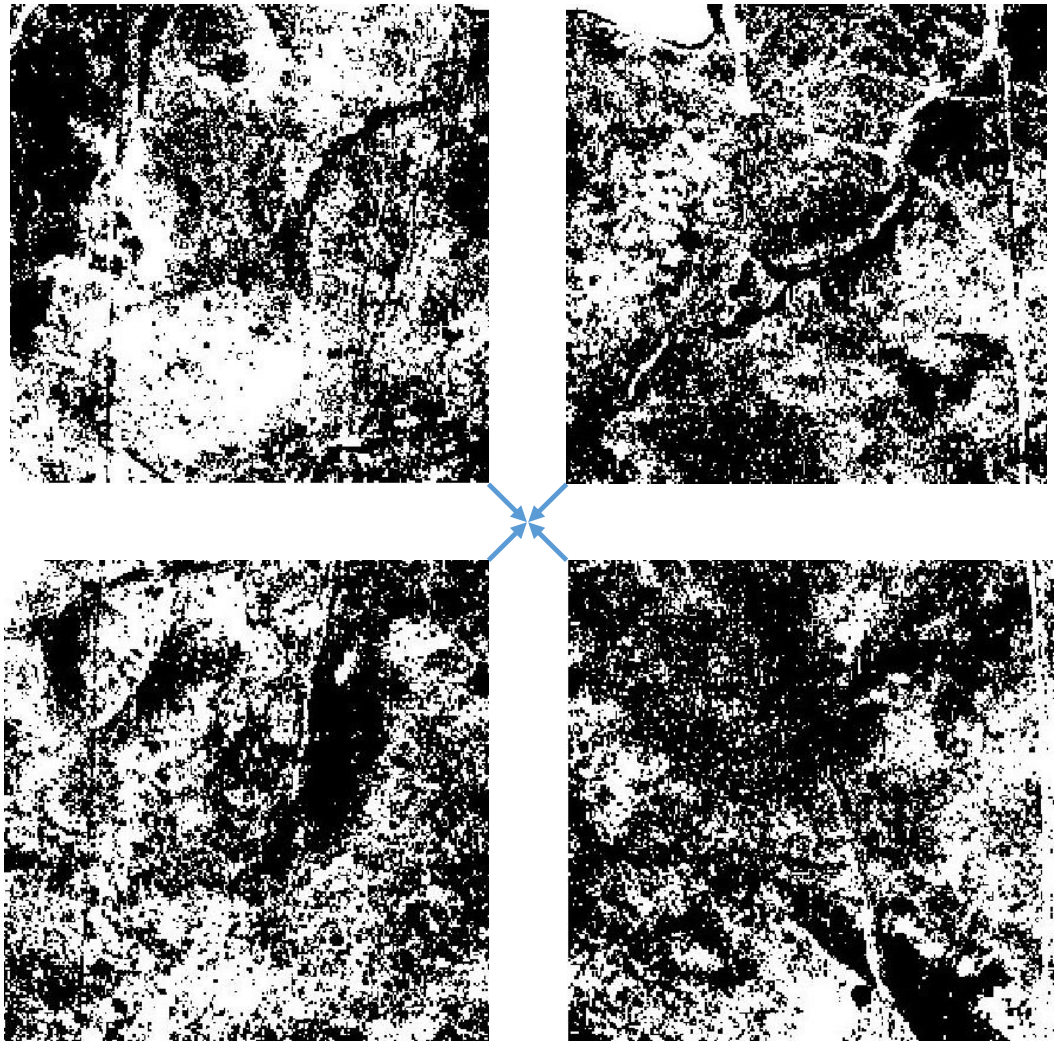
Use Multiprocessing Module

run_arc_parallel.py

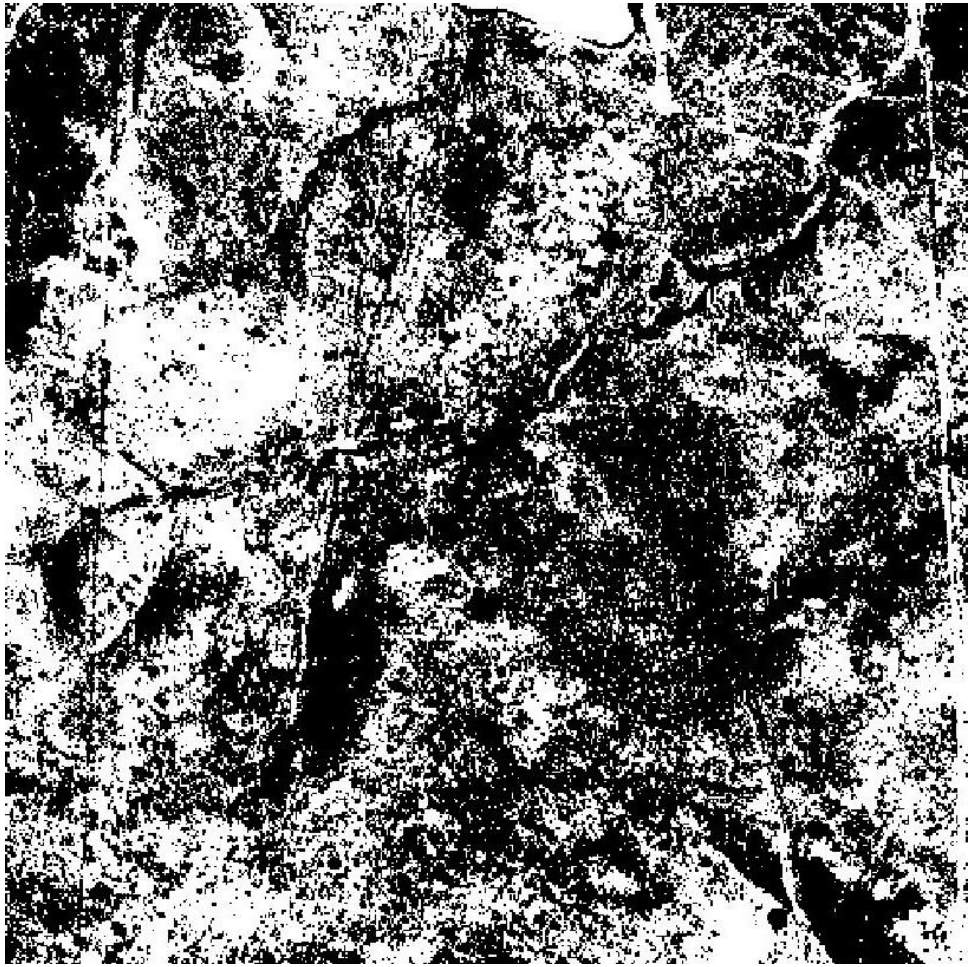


```
# Imports                                   from multiprocessing import Process
                                            import subprocess

# Function to run each process             def run_shell(command):
                                                p = subprocess.Popen(command)
                                                p.communicate()

                                            def main(argv):
                                                for each part:
# Create process                                    command = "python process_arc_part.py " + argsStr
                                                    task = Process(target=run_shell, args=(command,))
                                                    task.start()
                                                    tasks.append(task)

# Wait for all processes to finish              for task in tasks:
                                                    task.join()
```

Append Output Parts



Return Full Output Rasters



Review Performance Considerations

- Sample run
  - 4 input rasters, 800 MB each
  - 4 Con calls
  - Single run, Apply Raster Conditions tool - 6.5 minutes
  - Parallel run
    - Splitting - 25 minutes
    - 4 parts, Apply Raster Conditions tool - 2.5 minutes
    - Appending - 1.5 minutes

- As number of Con statements increases
  - Relative cost of splitting decreases
  - Overall time savings increase

Consider Wider Applicability

- Processing requirements continually increasing

  E.g,
  - NAIP imagery improving from 3.5 ft to 1 ft resolution
  - LIDAR popularity growing

- Concept can be applied to any geoprocessing operation for which tasks can be separated into independent parts

Thank You for Coming!

- David Howes
  - David Howes, LLC, Seattle, WA
  - GIS tools, processes & supporting infrastructure
  - http://dhowes.com
  - david@dhowes.com

- Eric Sant
  - Open Range Consulting, Park City, UT
  - Rangeland management
  - http://newfoundgeo.com

For slides and other resources, please see:

- http://gispd.com/events
- http://www.dhowes.com/presentations