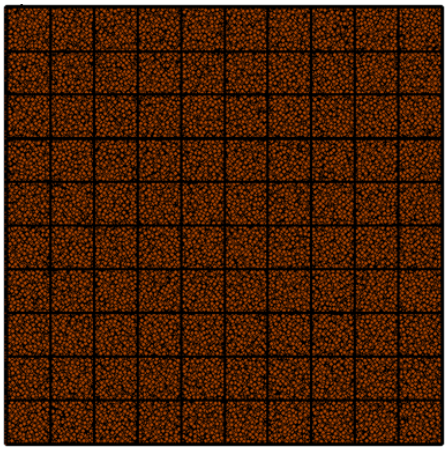
Lab: HP Geocomputing in Amazon Virtual Machine

Lab. High Performance Geocomputing on Amazon Virtual Machine

The multiprocessing Python module enables the distribution of computation work between multiple processes, taking advantage of multiple CPU cores and larger amounts of available system memory. The goal of this lab is to develop a Python script for processing vector GIS datasets with many features and to run the script on an Amazon Virtual Machine. The task is to find identical features among 1 million points and the idea is to improve the performance by splitting data into groups to be processed simultaneously. A fishnet of tiles is pre-created to define the spatial extents for the groups.



Python Script

The script consists of two major functions: **find_identical** and **main** . Find_identical is basically a worker function to perform finding identical within a given tile and returning a numpy array as the result. In the main function, a pool object is created and the find_identical function is mapped to a list of all the tiles to run the jobs in parallel.

a) Import the modules and set the environment

```
import multiprocessing
import numpy
import arcpy
arcpy.env.overwriteOutput = True
```

b) Find_identical function

```
def find_identical(oid):
    '''Worker function to perform Find Identical, and return'''
    '''a list of numpy arrays as the results.'''
    # Create a feature layer for the tile in the fishnet.
    tile = arcpy.management.MakeFeatureLayer(
                                'YOUR_WORKSPACE/fishnet',
                                'layer{0}'.format(oid[0]),
                                """OID = {0}""".format((oid[0])))

    # Get the extent of the feature layer and set the environment
    # to use during Find Identical.
    tile_row = arcpy.da.SearchCursor(tile, 'shape@')
    geometry = tile_row.next()[0]
    arcpy.env.extent = geometry.extent

    # Execute Find Identical
```

```
        identical_table = arcpy.management.FindIdentical(
                                        'YOUR_WORKSPACE/Points',
                                        'in_memory/identical', 'Shape')

        # Convert the resulting table into a numpy array and return
        result_array = arcpy.da.TableToNumPyArray(identical_table, ["*"])
        return result_array
        # End find_identical
```

c) main function

```
def main():
    t1 = time.time()
 #
 Cre
 ate
 a
 list
 of
 OI
 D's
 use
 d
 to
 chu
 nk
 the
 inp
 uts
        fishnet_rows = arcpy.SearchCursor(
                'YOUR_WORKSPACE/fishnet', '', '', 'OID')
        oids = [[row.getValue('OID')] for row in fishnet_rows]

        # Create a pool class and run the jobs--
        # the number of jobs is equal to the length of the oids list
        pool = multiprocessing.Pool()
        result_arrays = pool.map(find_identical, oids)

        # Concatenate the resulting arrays and create an output table
        # reporting any identical records.
        result_array = numpy.concatenate(result_arrays,axis=0)
        arcpy.da.NumPyArrayToTable(result_array,
                                        'YOUR_WORKSPACE/identical')

        # Synchronize the main process with the job processes to ensure
        # proper cleanup.
        pool.close()
        pool.join()

        t2 = time.time()

        # Print the time spent
        print t2 - t1
        # End main

if __name__ == '__main__':
```

main()

AWS Virtual Machine

Now you can log into your AWS account and launch four EC2 instances with 4, 8, 12, and 16 CPU cores, respectively. ArcGIS then needs to be installed on these EC2 virtual machines. The instruction for ArcGIS installation on Amazon EC2 can be found in this Youtube video. https://www.youtube.com/watch?v=fX_Cdh6DzQM . You can then run the script on these virtual machines and record the time spent on each. Plot out the time spent against the number of CPU cores to see how the performance has been improved.

Credit: the python script is modified from the example found at

http://blogs.esri.com/esri/arcgis/2011/08/29/multiprocessing/

Download Word Document Copy of Lab Assignment

Creator Name	Jie Tian
Content Title	HP Geocomputing in Amazon Virtual Machine
Content Type	Lab Assignment
Part Of	Module 3: High Performance Geospatial Computing
Learning Objectives	
Background Knowledge	
Resources Needed	
Work Mode	
Relation to Project	
Feedback Needed	