Distributed Image Analysis Using the ArcGIS API for Python

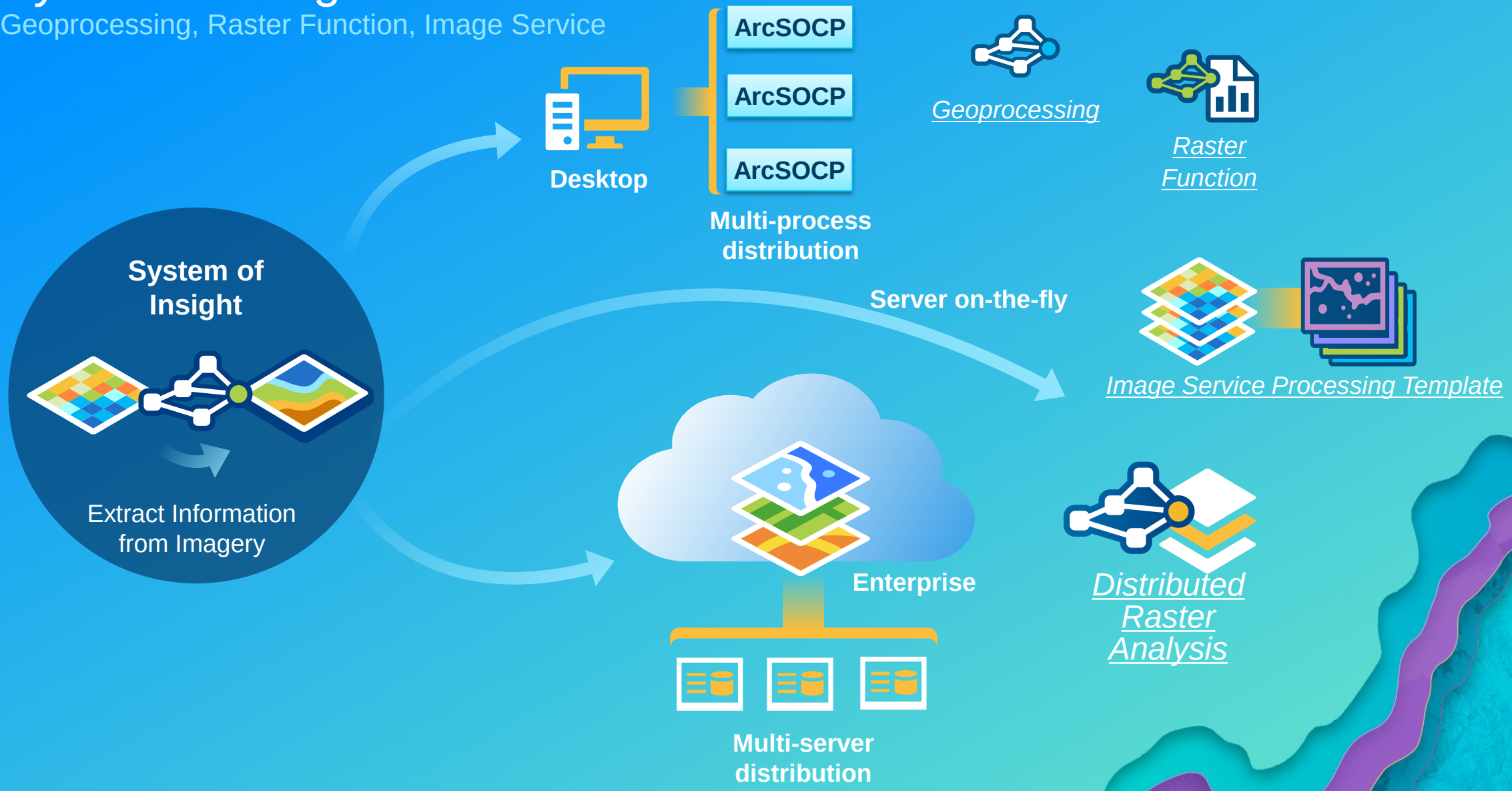
Jie Zhang, Elizabeth Graham

The ArcGIS Platform

Is a Comprehensive Imagery Platform

System of Insight

Geoprocessing, Raster Function, Image Service



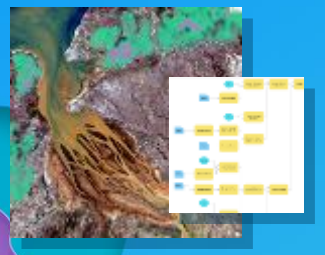
Enterprise Image Server with Distributed Raster Analysis

- ***From 10.5, ArcGIS has a new way to create and execute spatial analysis models and image processing chains which leverage distributed storage and analytics***
  - ***Raster Analysis works with your existing GIS data and imagery***
    - *register your data with Image Server without converting*
  - ***Raster Analysis can optimize your data for distributed analytics***
    - *result imagery is written into distributed raster storage for improved scalability*
  - ***Raster Analysis is designed to scale with your organization's demands***
    - *scale up to get the job done quicker, scale down when resources are not needed*

Enterprise Image Server with Distributed Raster Analysis

- **Complete analysis task for data too big for single desktop machine**
  - **Meet time constraint (*Month -> Weeks -> Days -> Hours -> Minutes*)**

Dynamic Raster Models

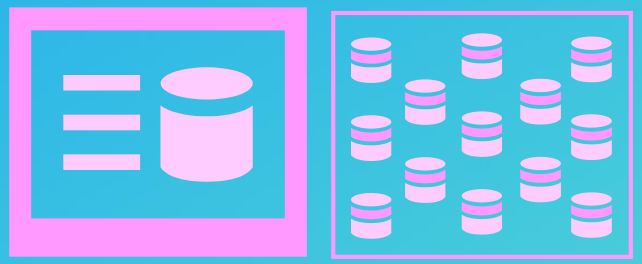


on-the-fly processing

Geoprocessing Models

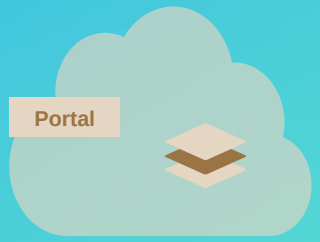
powerful analytics

Server-based Distributed Raster Analytics with Distributed Raster Data Storage



(persistent) **distributed analytics with optional distributed storage for even greater scalability**

Web GIS Layers



rich geoinformation model

Raster Analysis Capabilities

Large Collection of Raster Functions

Multiband Math

Arithmetic
Band Arithmetic

Math * *

Calculator
Abs
Divide
Exp
Exp10
Exp2
Float
Int
Ln, Log10
Log2
Minus
Mod
Negate
Plus
Power
Round Down
Round Up
Square
Square Root
Times

ACos
ACosH
ASin
ASinH
ATan
ATan2
ATanH
Cos
CosH
Sin
SinH
Tan
TanH

Con
Set Null

Bitwise
And
Left Shift
Not
Or
Right Shift
Xor
Boolean
And
Not
Or
Xor
Equal To
Greater Than
Greater Than Equal
Is Null
Less Than
Less Than Equal
Not Equal

Correction

Apparent Reflectance
Geometric Correction
Speckle Filtering (Lee,Frost,Kuan)
Thermal noise *
Radiometric Calibration *

Data Management & Conversion

Raster to Vector
Vector to Raster
Colormap
Colormap To RGB
Complex
Grayscale
Remap / Reclass
Spectral Conversion
Unit Conversion
Vector Field
LAS to Raster
LAS Dataset to Raster
Clip
Composite
Extract Bands
Mask
Mosaic Rasters
Rasterize Features
Reproject
Nibble * *

Visualization & Appearance

Contrast and Brightness
Convolution
Pansharpening
Resample
Statistics and Histogram
Stretch

Interpolation

Interpolate Irregular Data
- Nearest Neighbor
- IDW
- EBK
Swath

Surface Generation & Analysis

Aspect
Curvature
Elevation Void Fill
Hillshade
Shaded Relief
Slope
Viewshed *

Analysis: Distance & Density *

Euclidean Distance
Cost Distance
Least Cost Path
Kernel Density

Analysis: Overlay

Weighted Overlay *
Weighted Sum * *

Analysis: Band Math & Indices

NDVI / NDVI Colorized
SAVI / MSAVI / TSAVI
GEMI
GVI (Landsat TM)
PVI
Tasseled Cap (Kauth-Thomas)
Binary Thresholding

Statistics:

Zonal Statistics * *
Cell Statistics * *
ArgStatistics

Analysis: Image Segmentation & Classification * *

Segmentation (Mean Shift)
Training (ISO, SVM, ML)
Supervised Classification

Analysis: Hydrology * *

Fill
Flow Accumulation
Flow Direction
Flow Distance
Stream Link
Watershed

Python

Custom Algorithms

*** New in 10.6**
*** Requires Spatial Analyst Ext. in Pro** or
*** Requires Image Analyst Ext.in Pro**
All functions are accessible on Image Server

Chain functions together into Raster Models and apply them to answer complex questions

ArcGIS Python API

ArcGIS Python API overview

- The ArcGIS API for Python is implemented using the online and on-premises web GIS platform provided by ArcGIS Online and ArcGIS Enterprise respectively.
- Distributed as the ***arcgis*** *conda* package.
- ***arcgis.raster*** module
  - The ***arcgis.raster*** module contains classes and **raster analysis functions** for working with raster data and imagery layers in ArcGIS Enterprise

arcgis.raster module

ImageryLayer

RasterCatalogItem

RasterManager

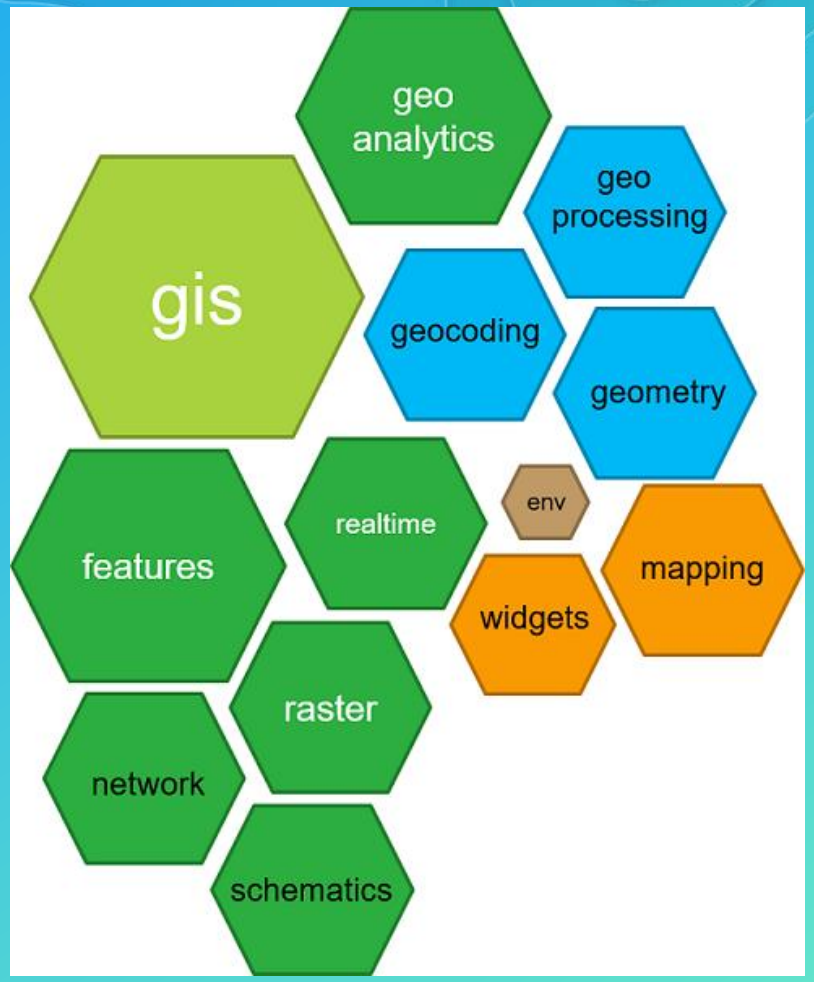
ImageryTileManager

Submodules

arcgis.raster.analytics module

arcgis.raster.functions module

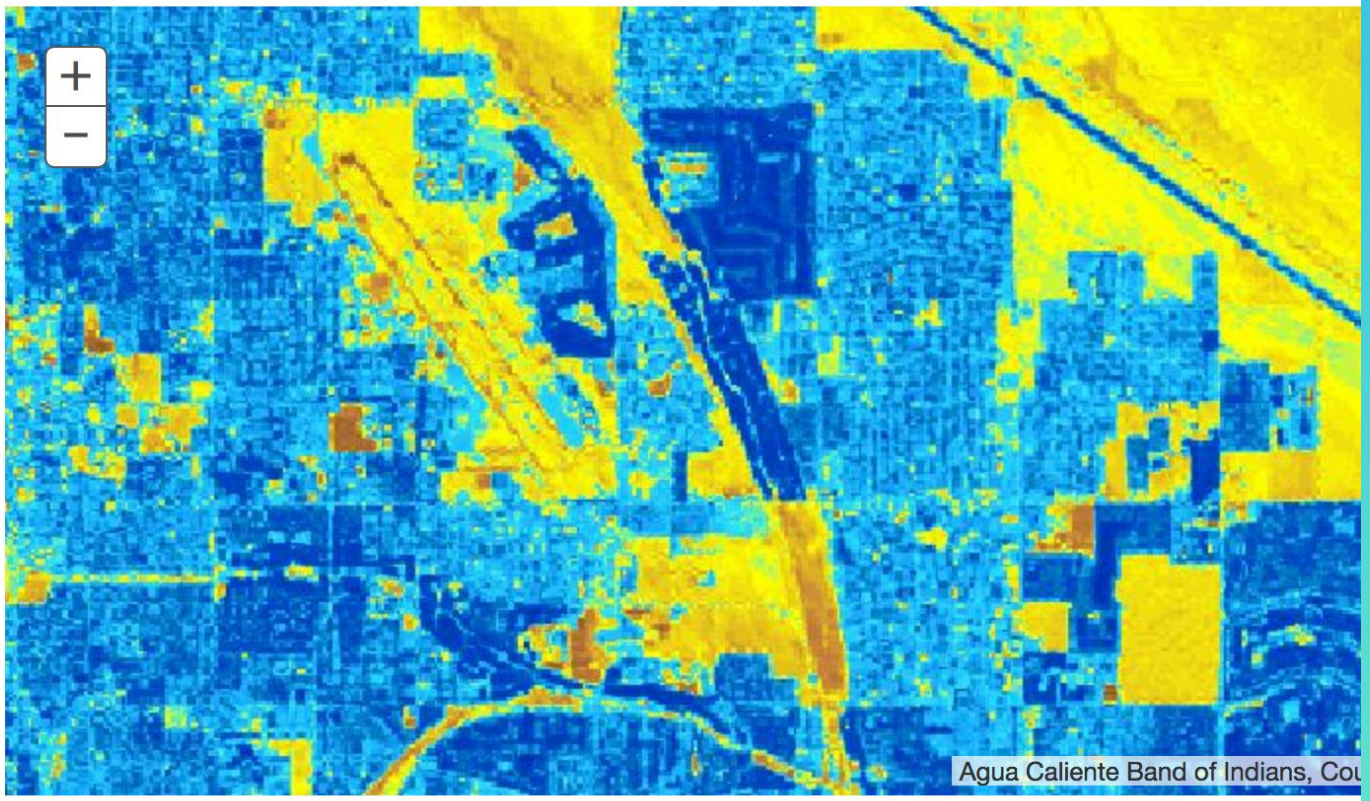
arcgis.raster.functions.gbl module



ArcGIS Python API for Raster Analysis - Image Layer

- Display Image Layer
- Apply Image Service on-the-fly processing

```
In [8]: map = gis.map("Palm Spring", zoomlevel=13)
        map
```




```
In [9]: map.add_layer(apply(l8layer, 'Normalized Difference Moisture Index Colorized'))
```

ArcGIS Python API for Raster Analysis – Raster Function

- Construct custom raster function chain
- Preview analysis result
- Submit raster analysis task

In [7]:
```
# Construct raster analysis workflow using raster function
# 1. Create slope and aspect raster from DEM
elevslope = slope(elev, slope_type="DEGREE", z_factor=1)
elevaspect = aspect(elev)
```

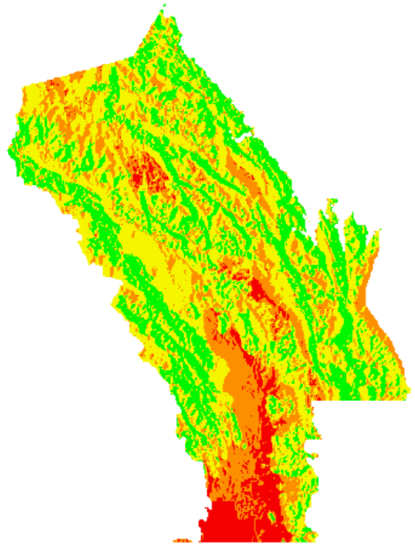
In [8]:
```
# 2. Apply suitability condition on slope raster (5 < slope < 60)
sslope = boolean_and([greater_than([elevslope, 5]), less_than([elevslope, 60])])
```

In [9]:
```
# 3. Apply suitability condition on aspect raster (0 < aspect < 200)
saspect = boolean_and([greater_than([elevaspect, 0]), less_than([elevaspect, 200])])
```

In [10]:
```
# 4. Apply suitability condition on DEM (30 < DEM < 400)
selev = boolean_and([greater_than([elev, 30]), less_than([elev, 400])])
```

In [11]:
```
# 5. Sum up using map algebra
output_values = [0, 64, 128, 255]
result = colormap(remap(sslope + saspect + selev,
                    input_ranges = [0,0,1,1,2,2,3,3],
                    output_values = output_values),
                    colorramp="Red To Green")
```

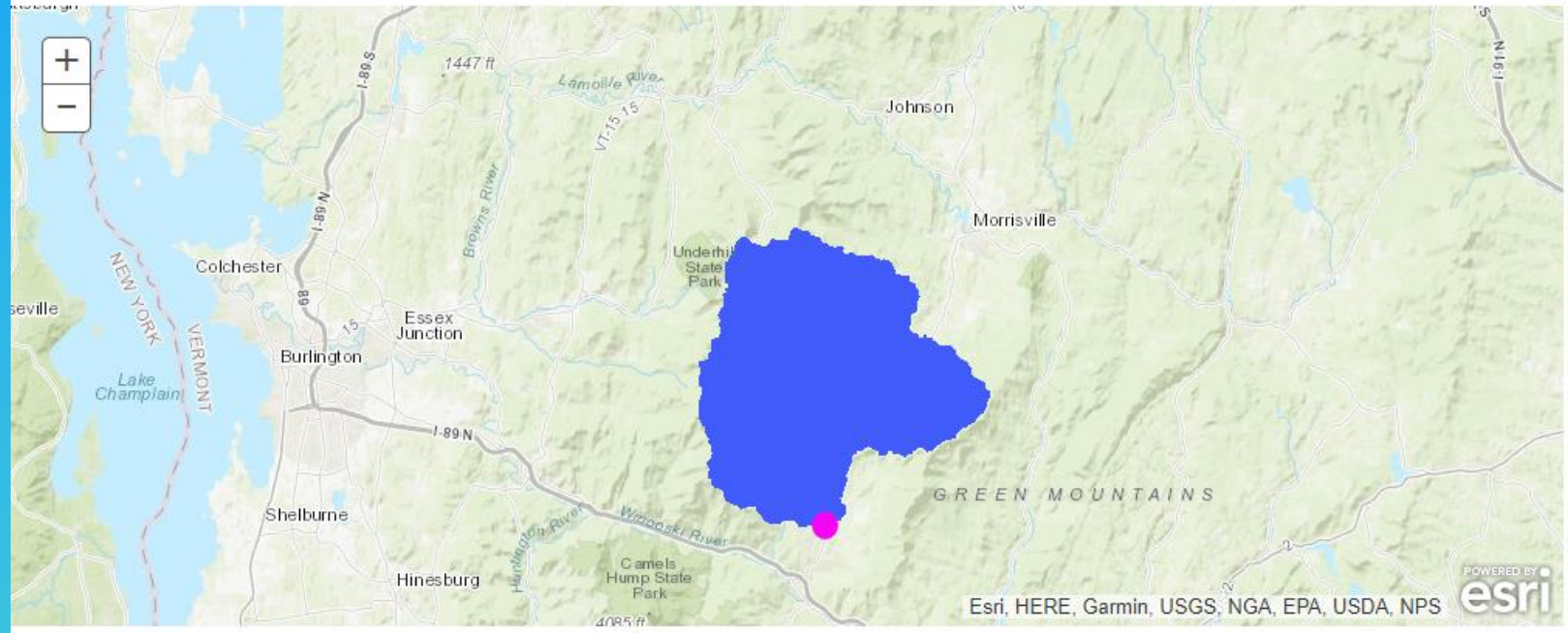
Out[11]:



ArcGIS Python API for Raster Analysis – Hydrology Workflow

- Global Analysis function generates persisted output before the next task

```
watershed_pp = enterprise.content.search("pourpoint")[0]
pp_water = watershed_pp.layers[0]
watershed_out = gbl.watershed(flow_dir_out, pp_water)
watershed_out_im = watershed_out.save("watershed_result")
watershed_out_lyr = watershed_out_im.layers[0]
imagery_map2 = enterprise.map("Stowe, VT", zoomlevel=10)
imagery_map2.add_layer(watershed_out_lyr)
imagery_map2
```



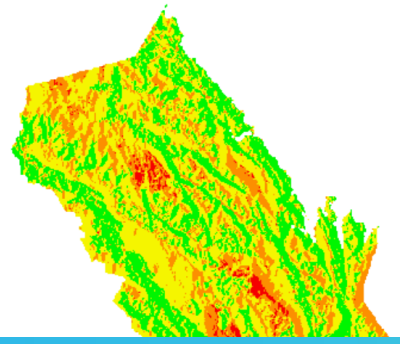
Raster Analysis using Python API

In [9]:
```
# 3. Apply suitability condition on aspect raster (0 < aspect < 200)
saspect = boolean_and([greater_than([elevaspect, 0]), less_than([elevaspect, 200])])
```

In [10]:
```
# 4. Apply suitability condition on DEM (30 < DEM < 400)
selev = boolean_and([greater_than([elev, 30]), less_than([elev, 400])])
```

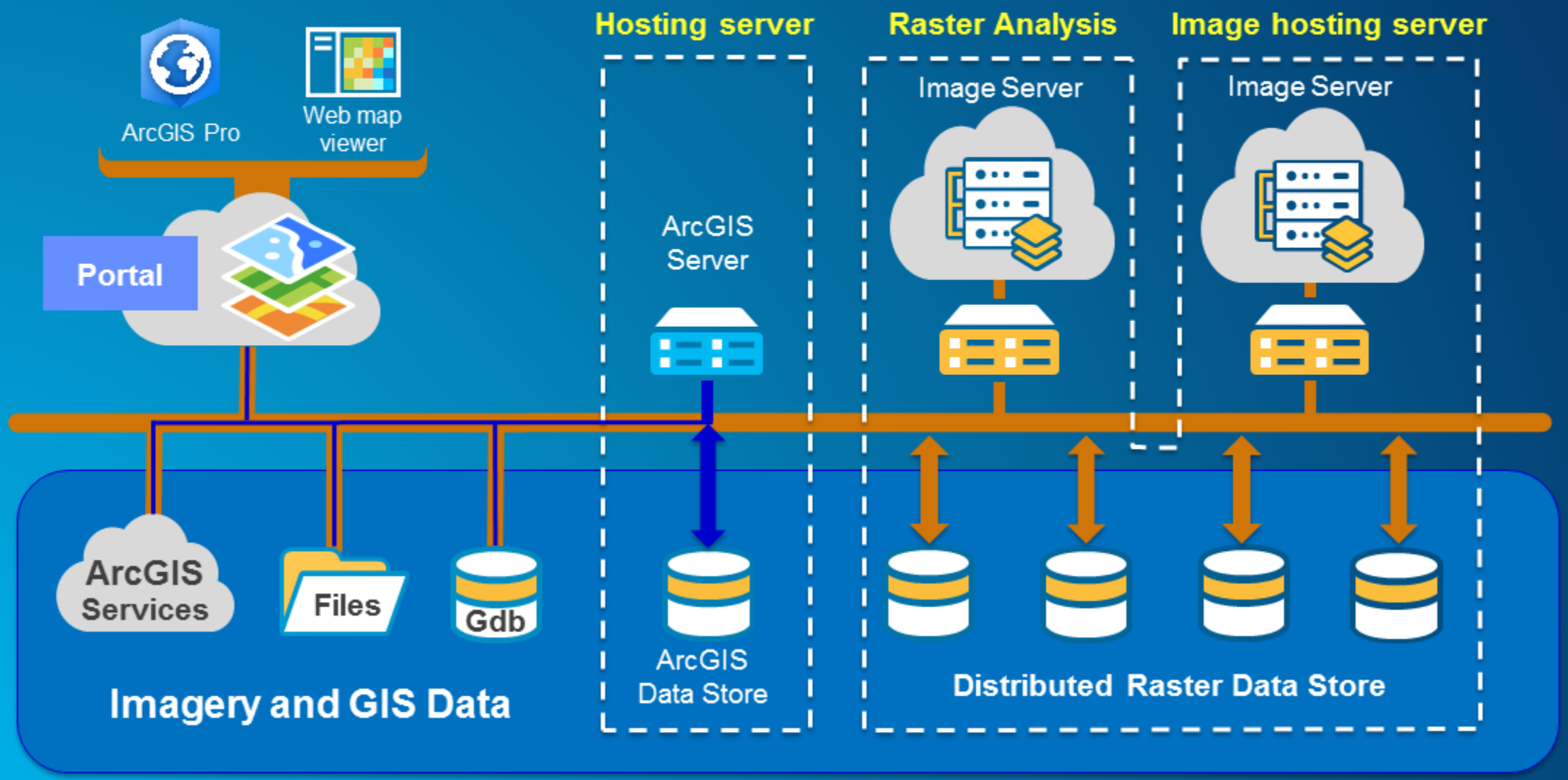
In [11]:
```
# 5. Sum up using map algebra
output_values = [0, 64, 128, 255]
result = colormap(remap(sslope + saspect + selev,
                    input_ranges = [0,0,1,1,2,2,3,3],
                    output_values = output_values),
                    colorramp="Red To Green")
```

Out[11]:



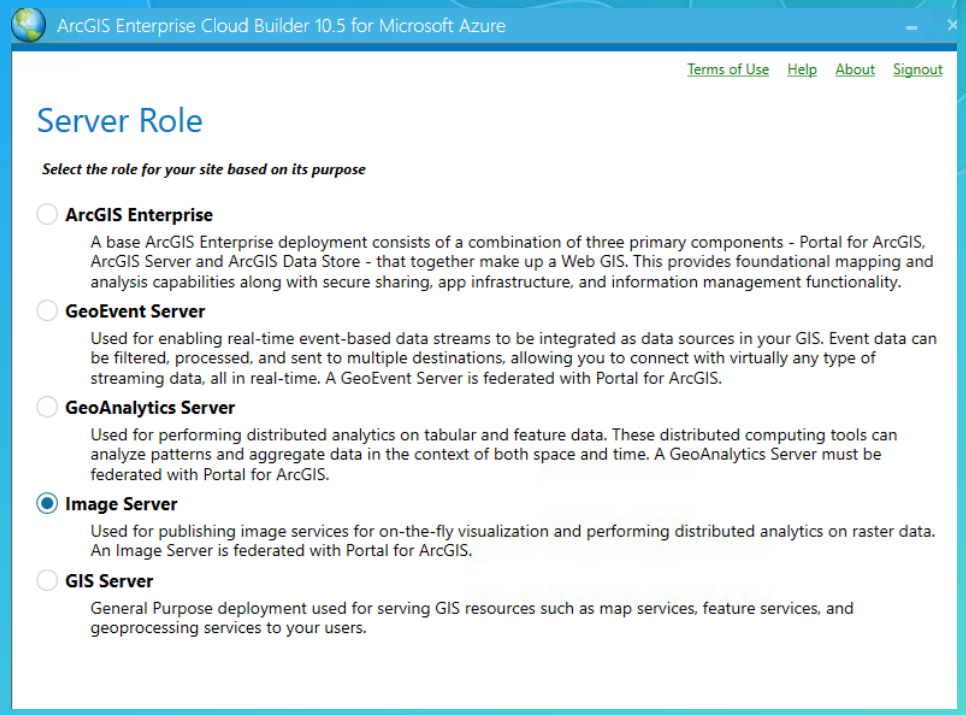
Raster Analysis System Deployment

Raster Analytics System Configuration



Deployment platform

- *On-premise + network file shared storage appliance*
- *Cloud platform + Cloud Object Storage*
  - *ArcGIS Cloud Builder for Azure*
  - *Amazon Cloud Formation template or Command line tool*



http://agsstore6.s3.amazonaws.com/8321/docs/index.html

Deploy additional ArcGIS Server deployments

In addition to the base ArcGIS Enterprise deployment, you can add functionality and capacity with additional ArcGIS Server sites.

Template Name	Description	Platform	View	Launch
ArcGIS GIS Server	This template creates a general purpose GIS Server site, or GeoAnalytics Server site, or RasterAnalytics Server site that you can federate with the portal in your base ArcGIS Enterprise deployment. You just need to use different Server license file to deploy different Server sites. ReadMe	Windows 2016	View	LAUNCH STACK
		Ubuntu 16.04 LTS	View	LAUNCH STACK

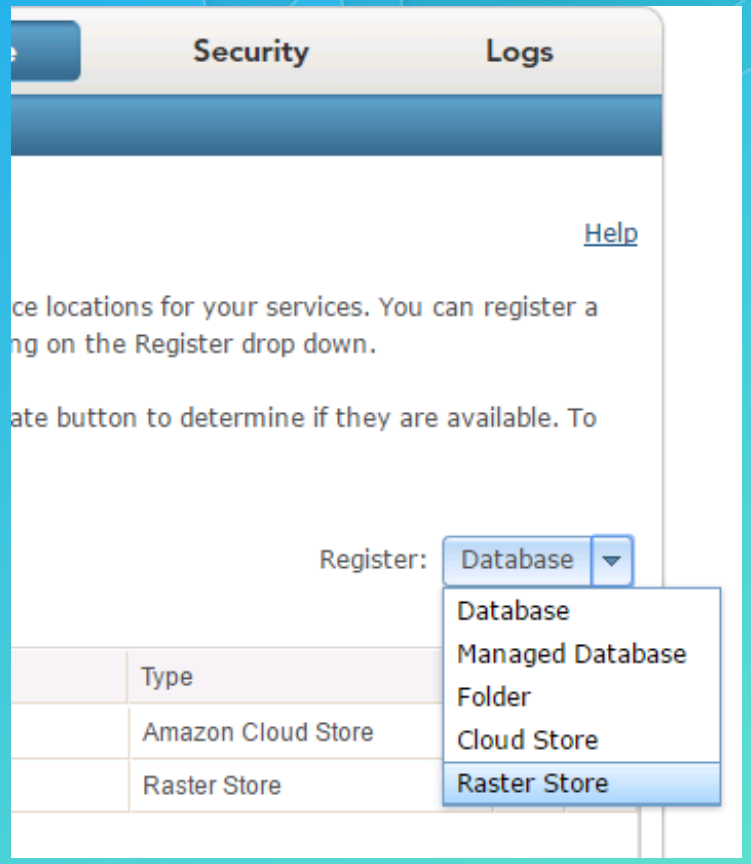
Raster Analysis System Components

Raster Store

- *New ArcGIS server data store type*
- *Raster Store registered to Image Server is for storing output imagery of Raster Analysis tasks*
  - *Shared file system storage*
  - *Cloud Object Storage*
    - *Amazon S3*
    - *Azure Blob Storage*

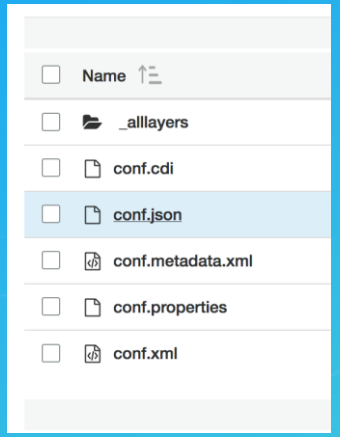
  Note: register a Cloud Store first then register a Raster Store reference to the Cloud Store

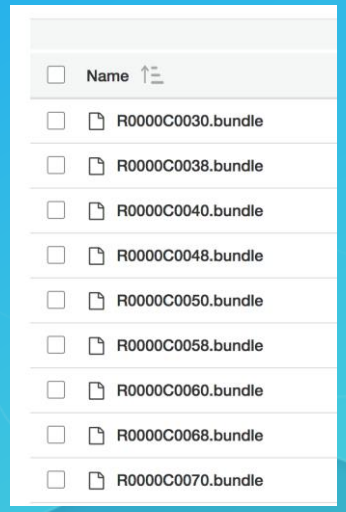
- *Feature output of Distributed Raster Analysis stored in Hosting server's*

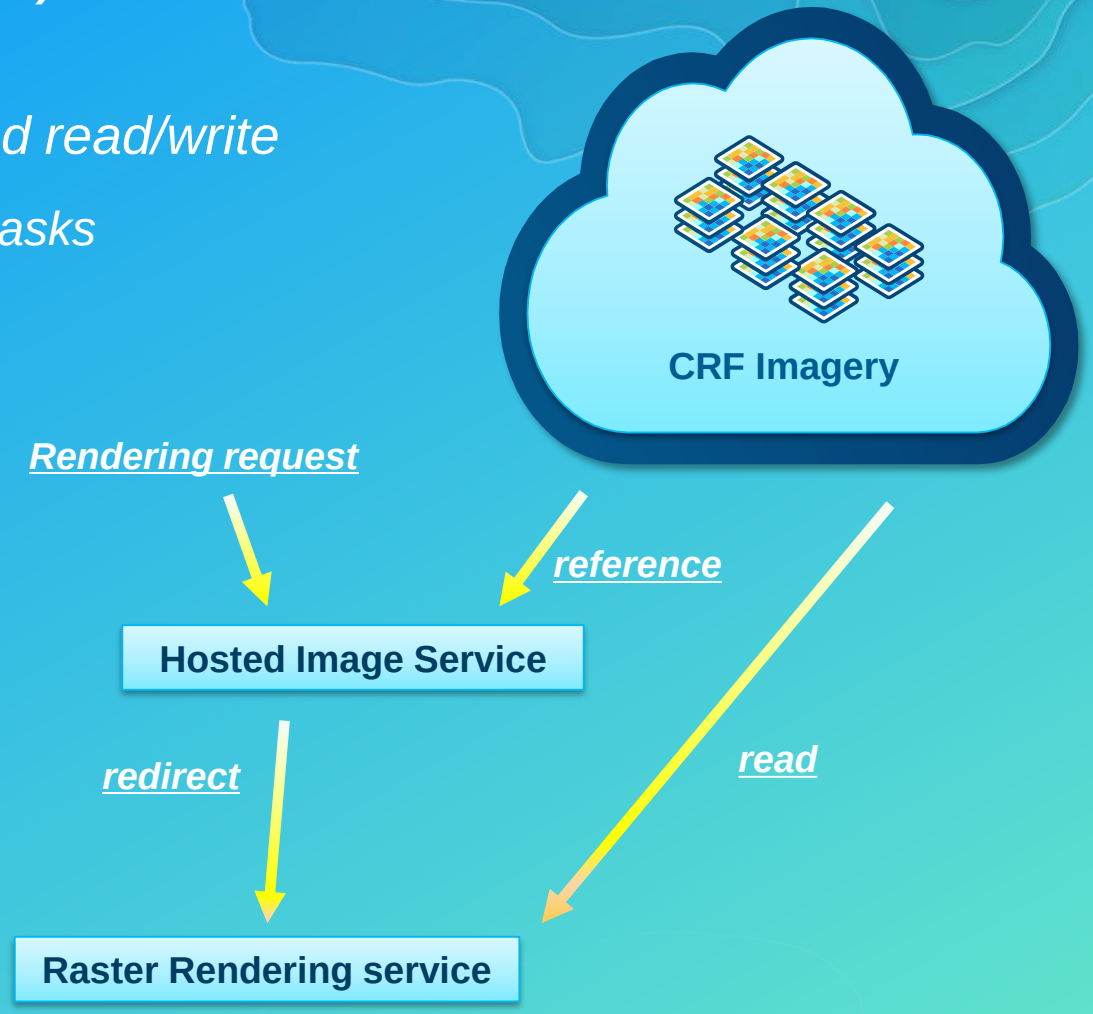


Optimized Cloud Raster Format (CRF)

- *New ArcGIS Raster Dataset format for distributed read/write*
  - *Default output imagery format for Raster Analysis tasks*
  - *Image is split to “bundles” with optimized schema*
  - *Support cloud storage and file system*
  - *Published as “hosted” image service*
  - *Rendered through Raster Rendering service*







Distributed Raster Analysis System Services

Geo Processing service as master

Image Service as worker

Image Service for rendering output

Raster Analysis Tools System Service

Folder: System

Current Version: 10.6

View Footprints In: ArcGIS Online Map Viewer

Services:

- System/CachingControllers (GPServer)
- System/CachingToolsEx (GPServer)
- System/CachingTools (GPServer)
- System/DistributedWorker (GPServer)
- System/PublishingToolsEx (GPServer)
- System/PublishingTools (GPServer)
- System/RasterAnalysisTools (GPServer)
- System/RasterProcessing (ImageServer)
- System/RasterRendering (ImageServer)
- System/ReportingTools (GPServer)
- System/SceneCachingControllers (GPServer)
- System/SceneCachingTools (GPServer)
- System/SyncToolsEx (GPServer)
- System/SyncTools (GPServer)
- System/UtilityNetworkTools (GPServer)

System/RasterAnalysisTools (GPServer)

Service Description: The RasterAnalysisTools service is used by

Tasks:

- GenerateRaster
- ConvertFeatureToRaster
- CopyRaster
- SummarizeRasterWithin
- ConvertRasterToFeature
- CalculateDensity
- CreateViewshed
- InterpolatePoints
- Classify
- Segment
- TrainClassifier
- GenerateRasterCollection
- StreamLink
- Watershed
- FlowDirection
- FlowAccumulation
- Fill
- DetermineOptimumTravelCostNetwork
- DetermineTravelCostPathsToDestinations
- CalculateDistance
- FlowDistance
- Nibble
- CalculateTravelCost

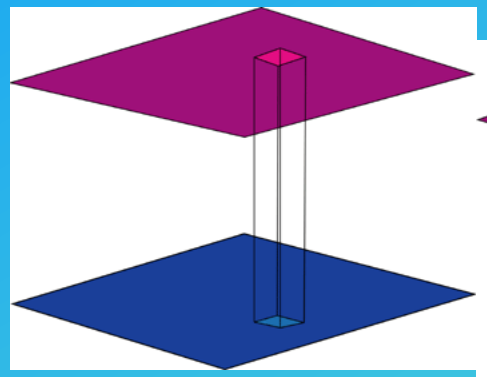
Note: *Generate Raster tool is the generic service tool takes raster function as input to support various analytic workflows*

ref: http://esriurl.com/rasterfunc

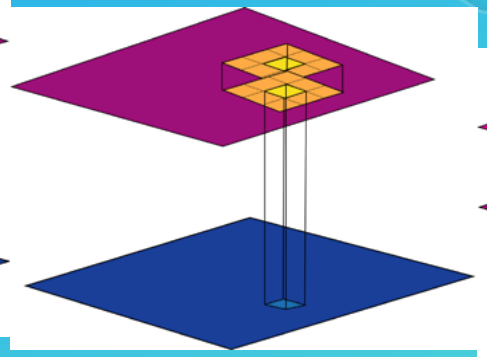
Types of Distributed Raster Analysis Operations

- Different type of operation poses different challenge to parallelization
- Distributed Raster Analysis supports
  - Local/Focal analysis
    - Most raster functions
    - Custom python raster functions
  - Zonal analysis
    - Zonal statistics
  - Global analysis
    - Hydrology Analysis
    - Cost Distance

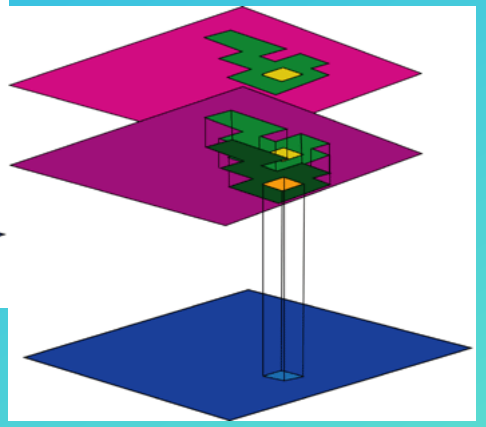
ref: http://esriurl.com/typeofops



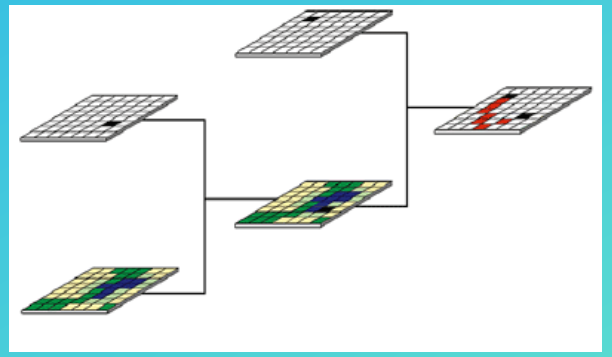
Local



Focal



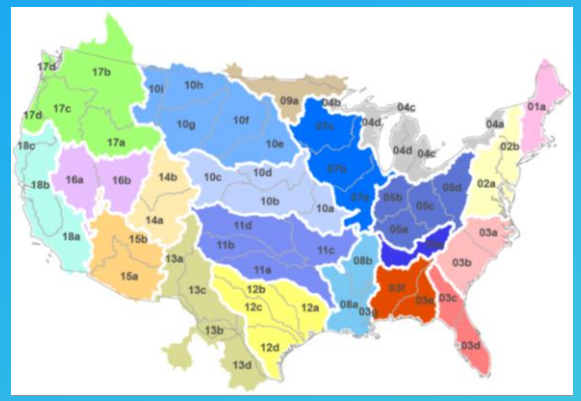
Zonal



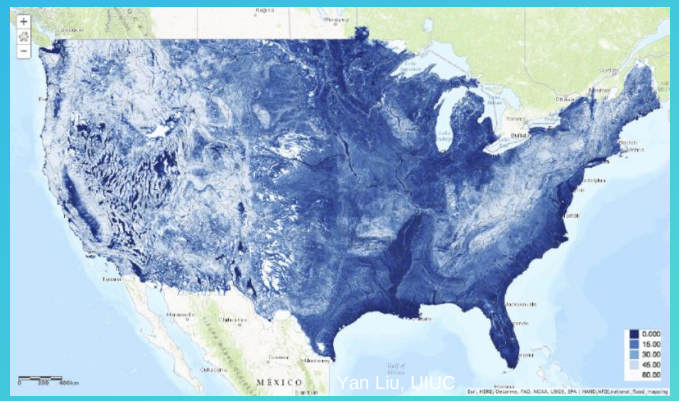
Global

Case Study: Hydro workflow scalability

- Computing hydrologic characteristics and flow patterns for US
  - For predicting stream flow and flood forecasting
- Old single thread tools required manual chunking of input
  - Performance suffered over half billion cells
- Now running 500x larger data on a distributed cluster



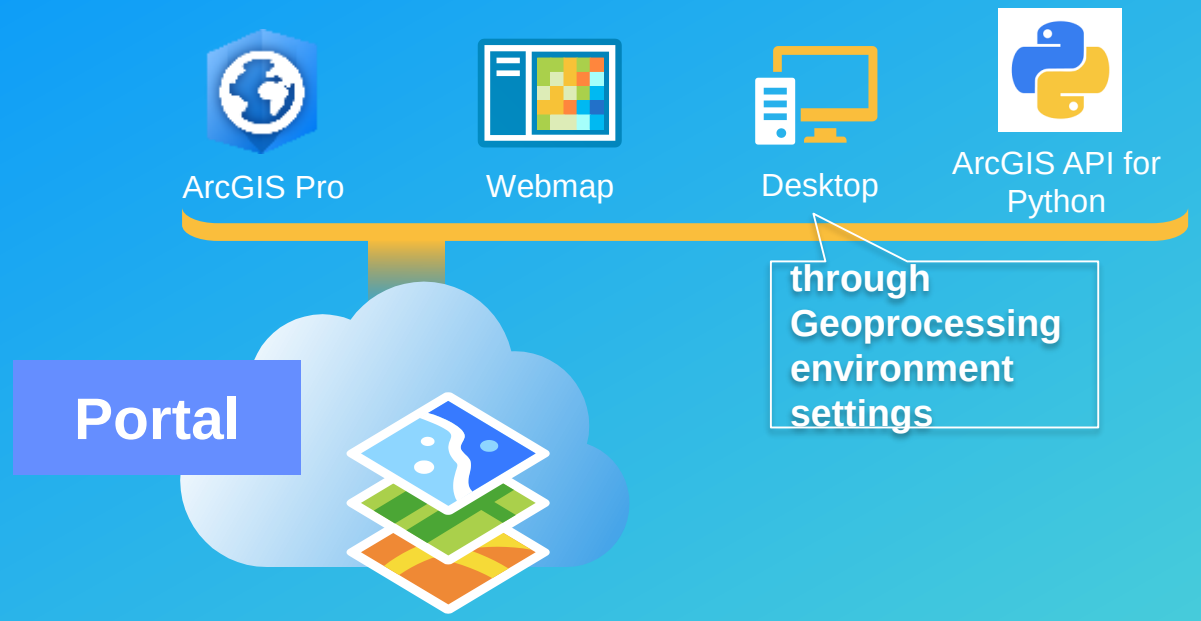
30m US – 25 Billion cells
57 separate processing units

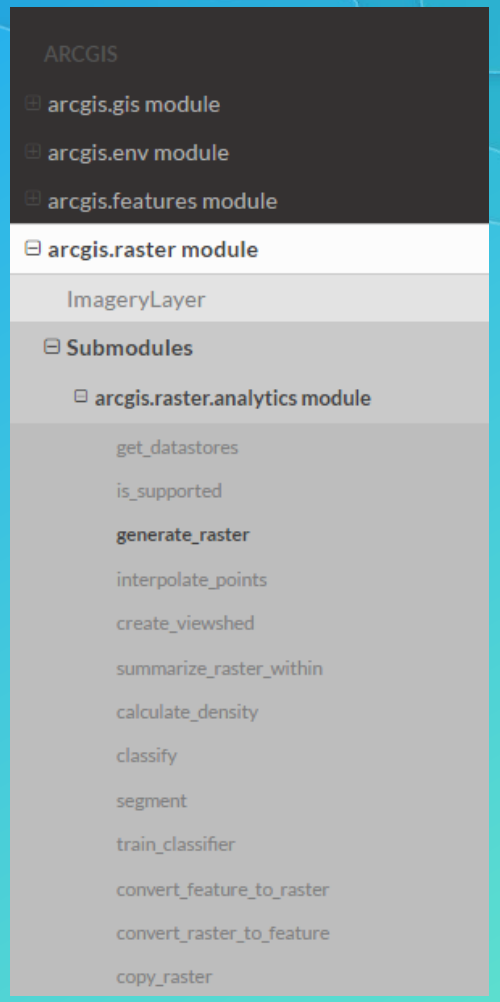


10m US – 230 Billion cells

Enterprise Raster Analysis Clients

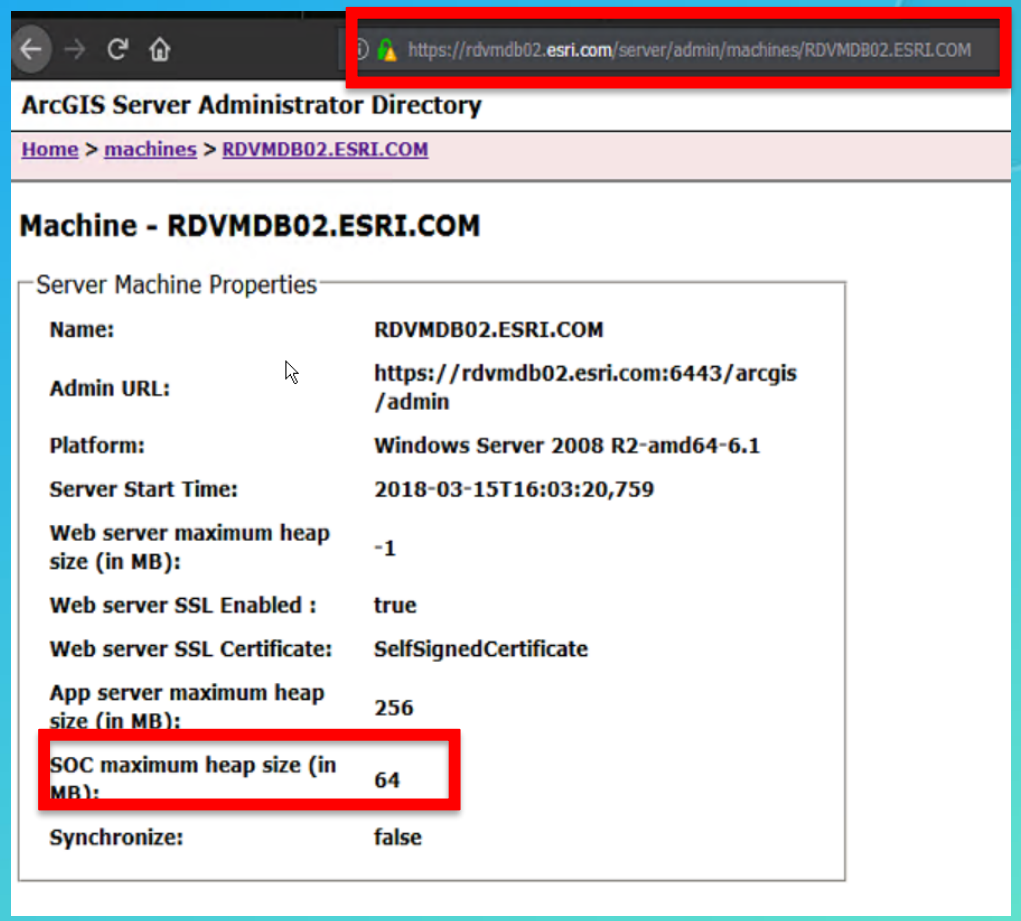
- ArcGIS Pro/Desktop
- Portal Webmap
- ArcGIS API for Python





Best practice, tips and tricks

- Split Raster Analytic and Image Hosting roles
- Increase server maximum heap memory for processing very large image
- Adjust maximum number of processing service instances based on machine specs
- A single raster analysis task is always default to use up to 80% of all the available raster processing service processes



esri

THE SCIENCE OF WHERE