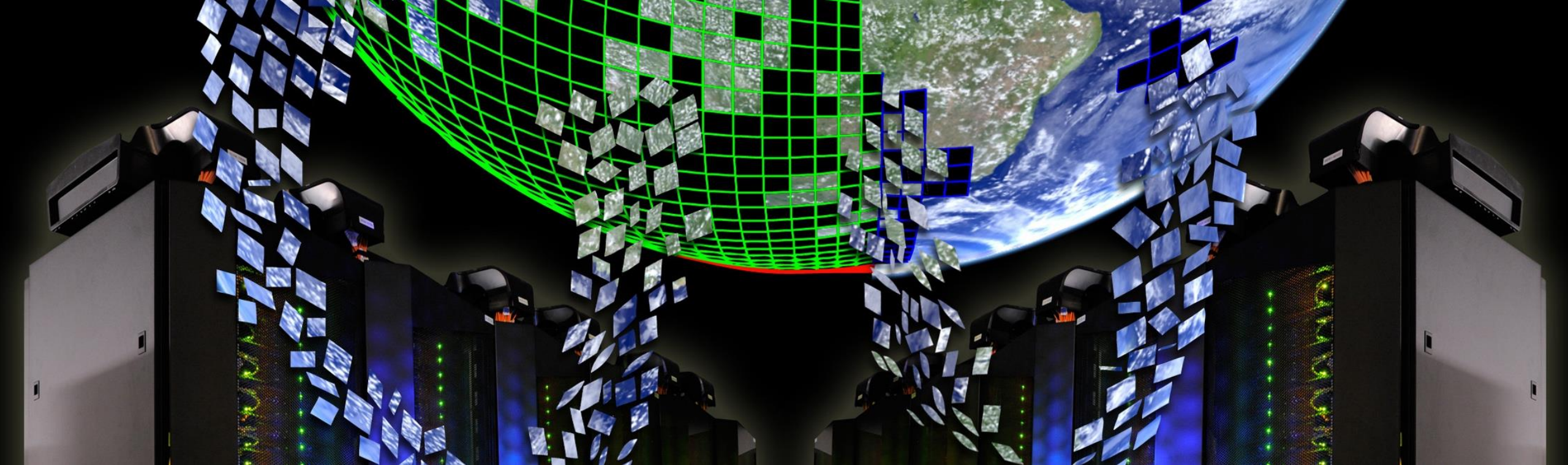




NASA Center for Climate Simulation (NCCS)

**Leveraging Python Multiprocessing to Manage a
Multi-Petabyte Raster Archive**

June 8, 2023

Jim Shute, Ryan Forbes, Erin Goodnough

- NCCS Mission
- NCCS Systems
- Functional Area Overviews
- GIS Platform Architecture
- NGA/Maxar/DigitalGlobe Data
- Ingest Process
- Challenges
- Development Workflow
- Standard Script Structure
- Demo
- Parallel Process Applicability and Results
- Multiprocessing Best Practices



The NCCS provides high performance computing for NASA-sponsored scientists and engineers. Our integrated set of computational capabilities includes High Performance Computing, Cloud Computing, Analytics, Data Sharing and Tools, Visualization, and Climate Data Services. The purpose of the NCCS is to enhance NASA capabilities in Earth science, with an emphasis on weather and climate prediction, and to enable future scientific discoveries that will benefit humankind.

Building 28, Goddard Space Flight Center

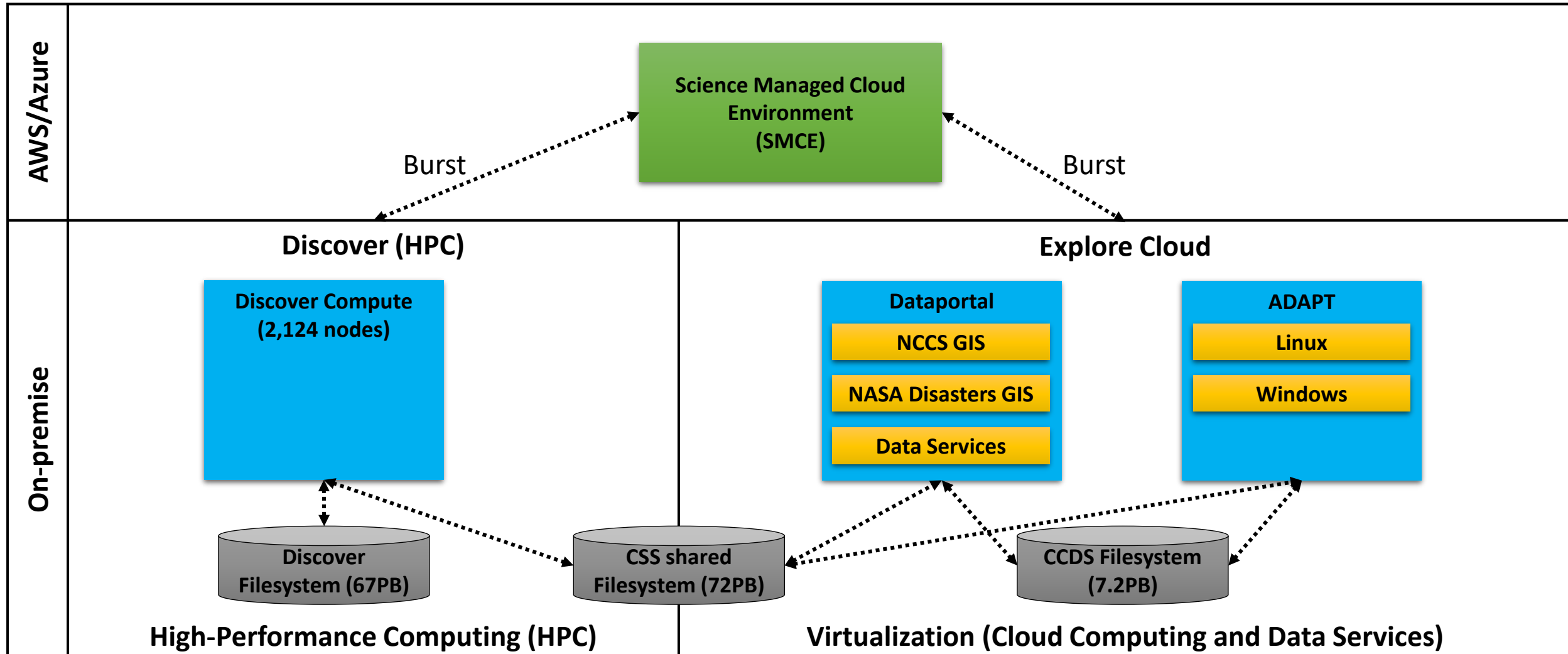


Piers J. Sellers Data Visualization Theater



ABOVE Science Team Meeting (ASTM5), 2019



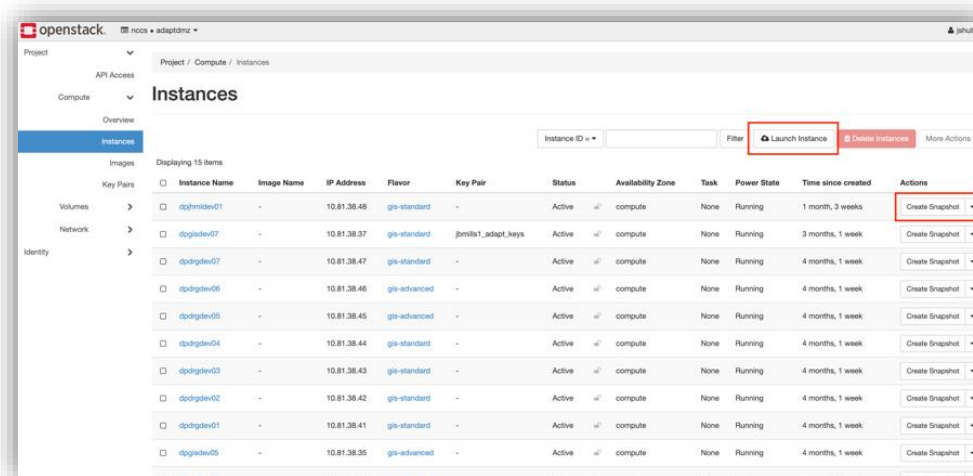


- Next generation on-premise cloud
- Multiple availability zones (B28, B28D, B32)
- Platform specifications
 - 275 hypervisors
 - 8,184 CPU cores
 - 68.8TB RAM
 - 7.2PB storage

Explore Cloud Control Plane – Building 28



Explore Cloud OpenStack Web Interface



- Centralized data store, accessible by all NCCS platforms/subsystems
- 72PB total storage

► **ABoVE: 80 TB**

- Oak Ridge National Laboratory Distributed Active Archive Center (**ORNL DAAC**) datasets, including over 130 archived as part of ABoVE.
- National Snow and Ice Data Center (**NSIDC**) DAAC datasets, including LVIS products archived in support of ABoVE.
- Alaska Satellite Facility (**ASF**) datasets related to L-band SAR.

► **AMSR-2: 5 TB**

► **AVHRR/Polar: 40 TB on ADAPT and 10 GB on CSS**

► **CFHA: 250 TB**

► **CMIP5: 105 TB**

► **CREATE-IP: 79 TB**

► **CSDA-Spire: 30 TB**

Note: Access permission required, please [contact](#) NCCS support

► **DSCOVR: 72 TB (L1B)**

► **DSCOVR: 72 TB (L2_CLOUD_03)**

► **FLDAS: 40 TB**

► **GeoMIP: 14 TB**

► **Geostationary (GOES): Ingest starting now, planning for 1 PB**

► **GEOS-IT: 420 TB**

Note: Public access coming soon.

► **GEOS-5 Nature Runs (g5nr): 5 PB**

► **HIMAT Snow Reanalysis: 5 TB**

► **ICEBridge: 2 TB**

► **ICESat: 8 TB**

► **ICESat-2: 161 TB**

► **IMERG: 15 TB**

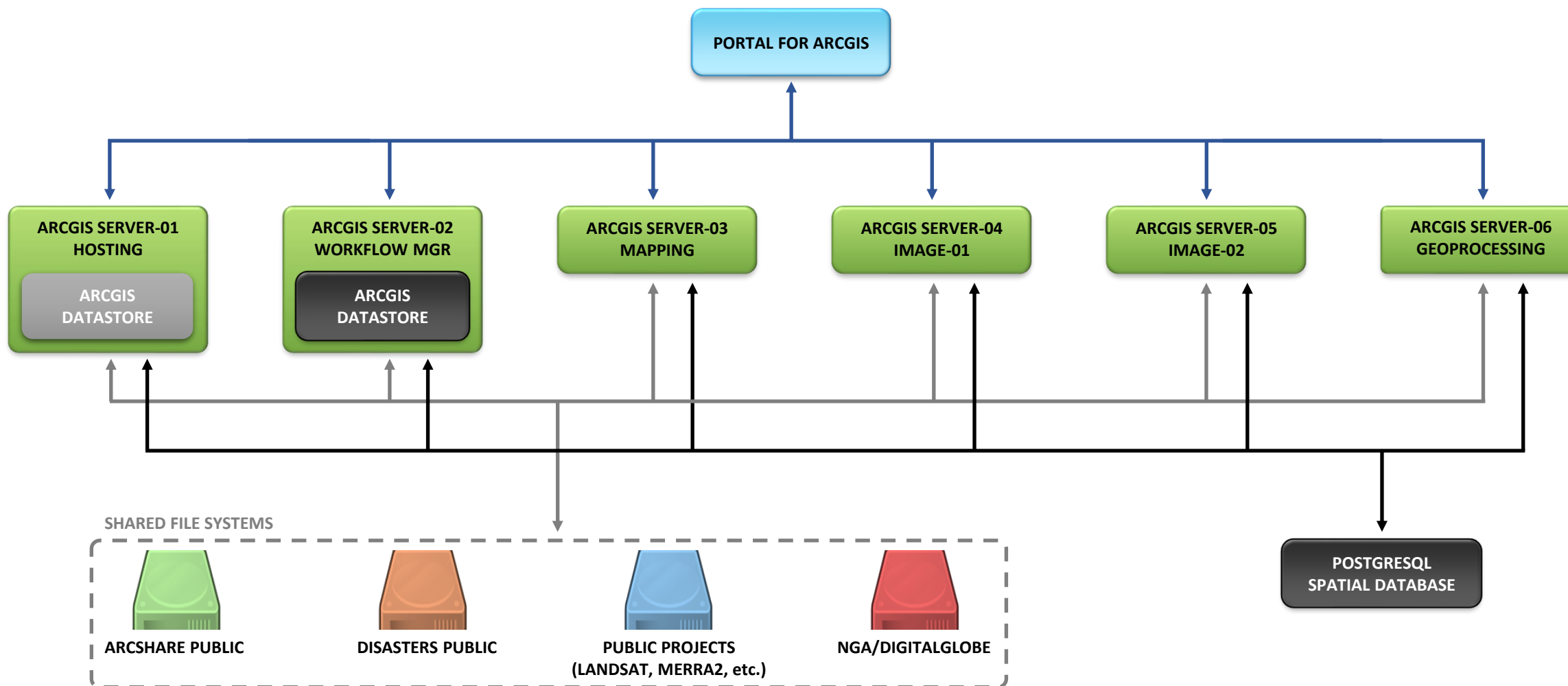
► **Landsat: 186 TB**

► **MAIAC: 107 TB**

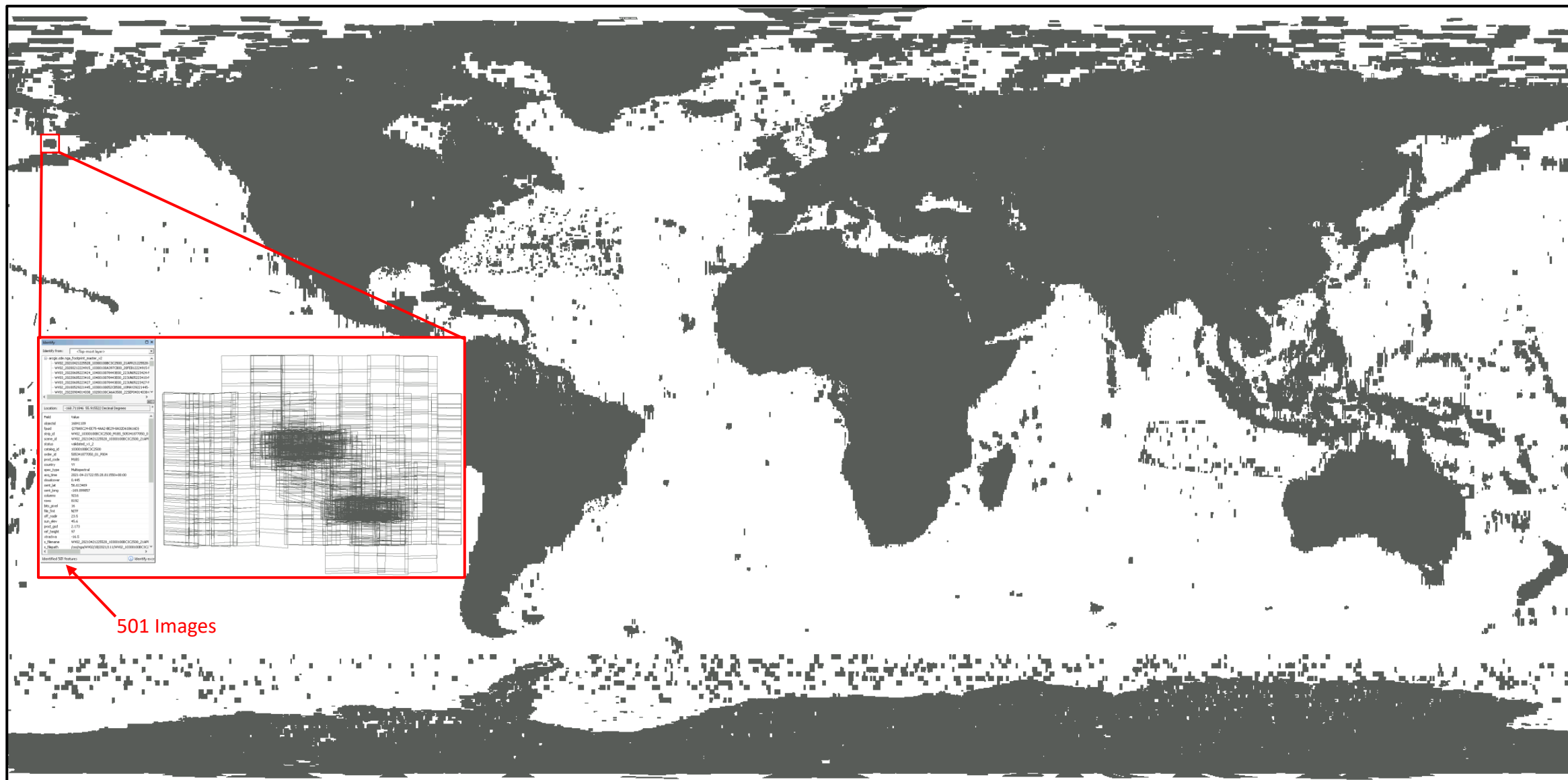
Subset of datasets stored on CSS

CSS1 and CSS2 (30PB total usable storage)

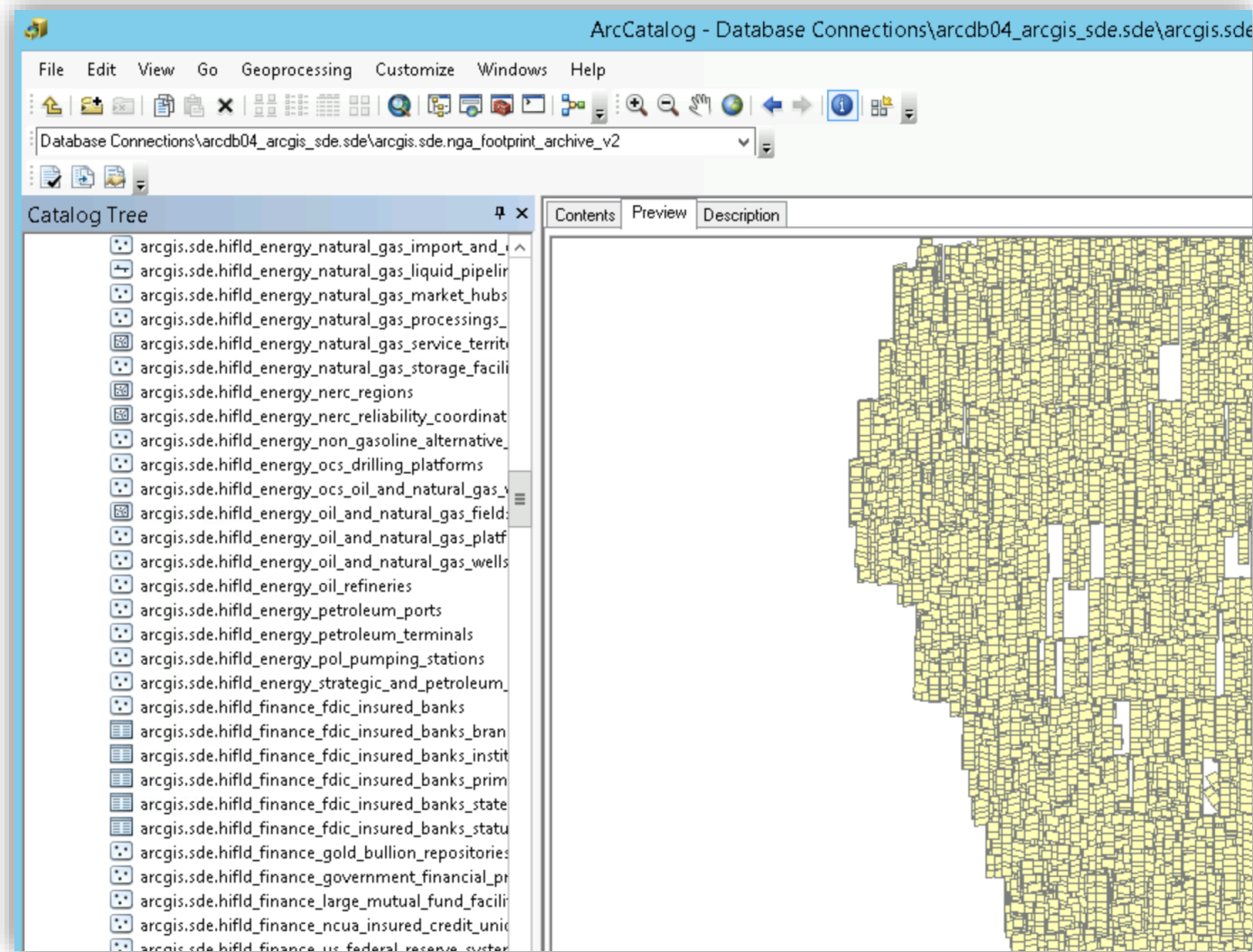




- 24.8 million images
- 14.5 petabytes
- Sensors
 - GE01 (2006859)
 - IK01 (320056)
 - OV03 (227)
 - QB02 (2304775)
 - WV01 (5353377)
 - WV02 (10392289)
 - WV03 (4416135)
 - WV04 (12455)
- Years
 - 1999 (10), 2000 (25,470), 2001 (34,011), 2002 (108,325), 2003 (172,904), 2004 (172,543), 2005 (177,619), 2006 (200,988), 2007 (270,816), 2008 (504,240), 2009 (600,408), 2010 (1,005,111), 2011 (1,477,337), 2012 (1,703,910), 2013 (1,592,369), 2014 (1,454,682), 2015 (1,612,178), 2016 (1,780,965), 2017 (1,837,000), 2018 (1,591,966), 2019 (1,382,450), 2020 (2,124,463), 2021 (2,337,951), 2022 (2,190,762), 2023 (447,695)



NGA/Maxar/DigitalGlobe Data Coverage



Ingest Process

FTP from DigitalGlobe



GridFTP from PGC



Hard drive

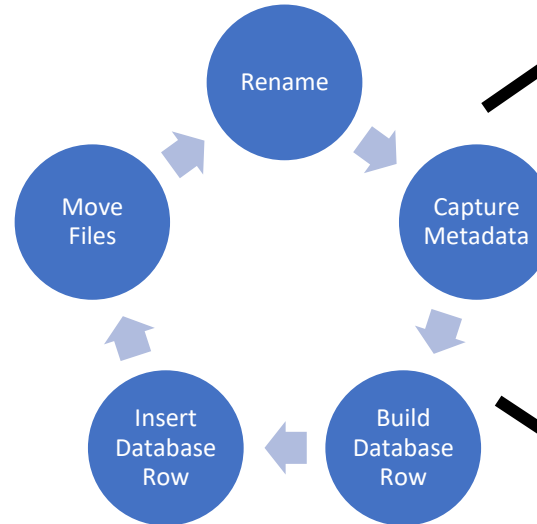


User Downloads



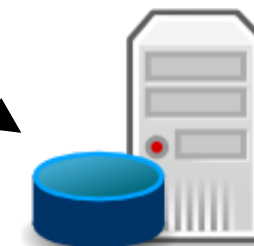
Initial File Location

Raster Ingest Process



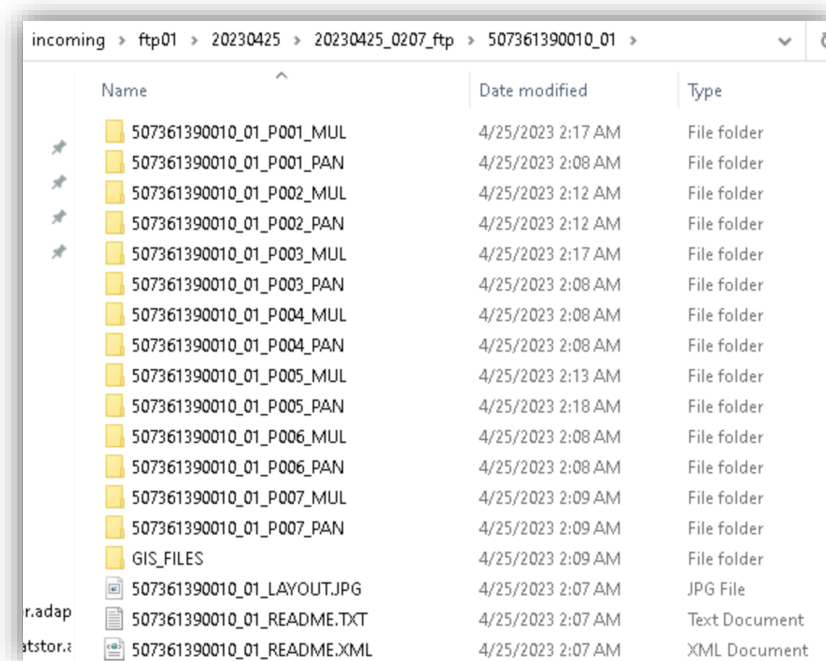
Final File Location

Raster Users

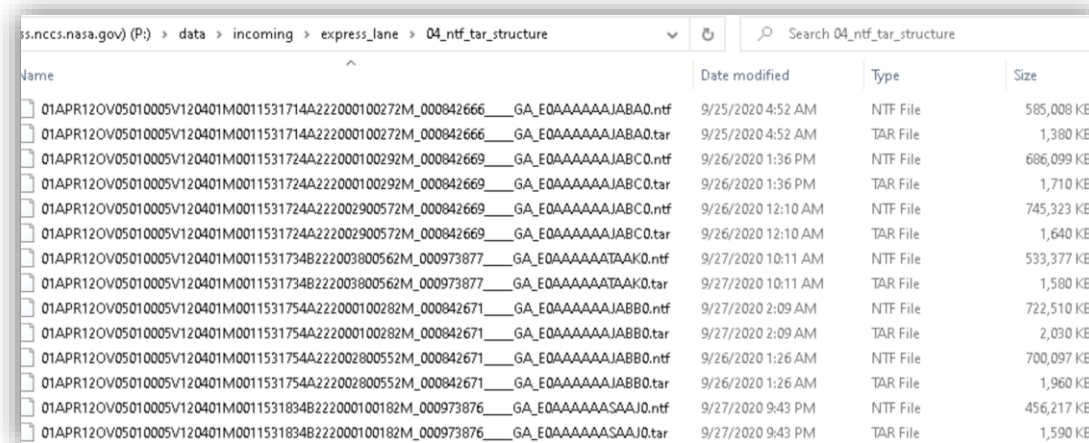


Footprints Database

- Multiple data streams and formats (e.g., “renamed” data vs. multiple raw file structures)
- At the beginning of the redesign process, image backlog was 1.1PB
- Slow ingest rates of 2,000 rows/hour
 - Backlog would have taken over 1 year
- Process and resulting database needed to meet the needs of multiple user groups
- Bad/duplicate data was complicating users’ raster analytics workflows

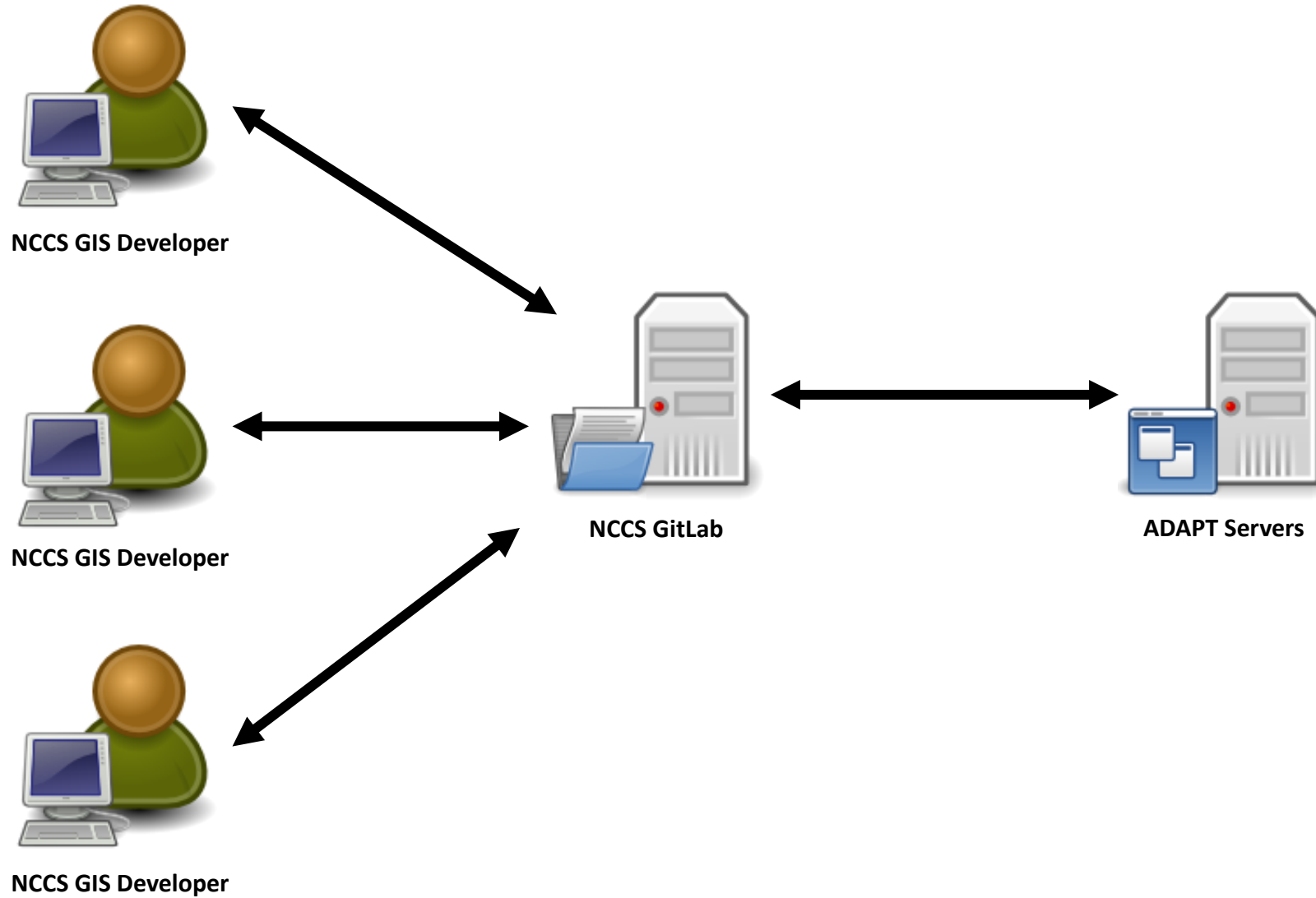


Name	Date modified	Type
507361390010_01_P001_MUL	4/25/2023 2:17 AM	File folder
507361390010_01_P001_PAN	4/25/2023 2:08 AM	File folder
507361390010_01_P002_MUL	4/25/2023 2:12 AM	File folder
507361390010_01_P002_PAN	4/25/2023 2:12 AM	File folder
507361390010_01_P003_MUL	4/25/2023 2:17 AM	File folder
507361390010_01_P003_PAN	4/25/2023 2:08 AM	File folder
507361390010_01_P004_MUL	4/25/2023 2:08 AM	File folder
507361390010_01_P004_PAN	4/25/2023 2:08 AM	File folder
507361390010_01_P005_MUL	4/25/2023 2:13 AM	File folder
507361390010_01_P005_PAN	4/25/2023 2:18 AM	File folder
507361390010_01_P006_MUL	4/25/2023 2:08 AM	File folder
507361390010_01_P006_PAN	4/25/2023 2:08 AM	File folder
507361390010_01_P007_MUL	4/25/2023 2:09 AM	File folder
507361390010_01_P007_PAN	4/25/2023 2:09 AM	File folder
GIS_FILES	4/25/2023 2:09 AM	File folder
507361390010_01_LAYOUT.JPG	4/25/2023 2:07 AM	JPG File
507361390010_01_README.TXT	4/25/2023 2:07 AM	Text Document
507361390010_01_README.XML	4/25/2023 2:07 AM	XML Document



Name	Date modified	Type	Size
01APR12OV05010005V120401M0011531714A222000100272M_000842666__GA_E0AAAAAAJABA0.ntf	9/25/2020 4:52 AM	NTF File	585,008 KB
01APR12OV05010005V120401M0011531714A222000100272M_000842666__GA_E0AAAAAAJABA0.tar	9/25/2020 4:52 AM	TAR File	1,380 KB
01APR12OV05010005V120401M0011531724A222000100292M_000842669__GA_E0AAAAAAJABCO.ntf	9/26/2020 1:36 PM	NTF File	686,099 KB
01APR12OV05010005V120401M0011531724A222000100292M_000842669__GA_E0AAAAAAJABCO.tar	9/26/2020 1:36 PM	TAR File	1,710 KB
01APR12OV05010005V120401M0011531724A222002900572M_000842669__GA_E0AAAAAAJABCO.ntf	9/26/2020 12:10 AM	NTF File	745,323 KB
01APR12OV05010005V120401M0011531724A222002900572M_000842669__GA_E0AAAAAAJABCO.tar	9/26/2020 12:10 AM	TAR File	1,640 KB
01APR12OV05010005V120401M0011531734B222003800562M_000973877__GA_E0AAAAATAAKO.ntf	9/27/2020 10:11 AM	NTF File	533,377 KB
01APR12OV05010005V120401M0011531734B222003800562M_000973877__GA_E0AAAAATAAKO.tar	9/27/2020 10:11 AM	TAR File	1,580 KB
01APR12OV05010005V120401M0011531754A222000100282M_000842671__GA_E0AAAAAAJABBO.ntf	9/27/2020 2:09 AM	NTF File	722,510 KB
01APR12OV05010005V120401M0011531754A222000100282M_000842671__GA_E0AAAAAAJABBO.tar	9/27/2020 2:09 AM	TAR File	2,030 KB
01APR12OV05010005V120401M0011531754A222002800552M_000842671__GA_E0AAAAAAJABBO.ntf	9/26/2020 1:26 AM	NTF File	700,097 KB
01APR12OV05010005V120401M0011531754A222002800552M_000842671__GA_E0AAAAAAJABBO.tar	9/26/2020 1:26 AM	TAR File	1,960 KB
01APR12OV05010005V120401M0011531834B222000100182M_000973876__GA_E0AAAAAASAAJO.ntf	9/27/2020 9:43 PM	NTF File	456,217 KB
01APR12OV05010005V120401M0011531834B222000100182M_000973876__GA_E0AAAAAASAAJO.tar	9/27/2020 9:43 PM	TAR File	1,590 KB

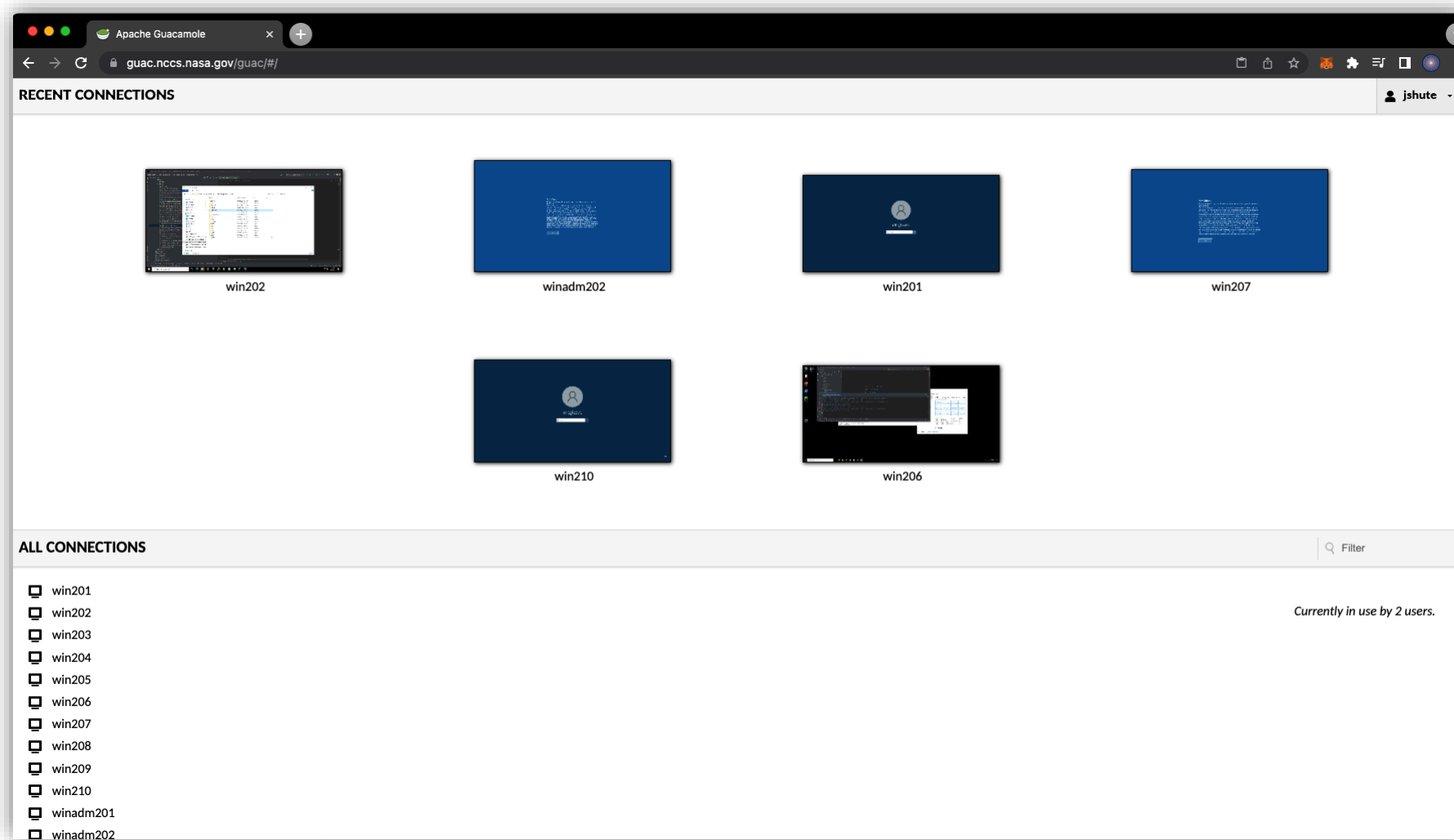
Development Workflow



1. Notes/purpose
2. Import modules
3. Configure global variables (if needed)
4. Supporting functions
5. Main function
 - a. Capture script parameters (if needed)
 - b. Import special script/class files (e.g., UtilitiesGeneral)
 - c. Global logger settings
 - d. Set up the main logger
 - e. Check script parameters
 - f. Configure pointers to subdirectories (e.g., output)
 - g. Import email recipients into a list
 - h. Set up counters
 - i. Set up list variables
 - j. Execute primary tasks (e.g., key parallel function)**
 - k. Process results
 - l. Calculate statistics
 - m. Prepare email content
 1. Attachments
 2. Email notification message body
 - n. Send email
 - o. Script complete message
6. Script start (executes the main function)

```

542 # Spatial Join and Calculate Attribute Function
543 # -----
544 def spatial_join_and_calculate_attribute(input_feature_class, join_feature_class, spatial_match_option,
545                                         join_attribute_field, source_calculate_field, target_attribute_field,
546                                         search_radius):...
589
590
591 # Split Path Function
592 # -----
593 def split_path(path):...
607
608
609 # Process Raster Strip Function
610 # -----
611 def process_raster_strip(strip_path, linux_path, windows_path, master_feature_class, master_feature_class_full_path,
612                           archive_feature_class, archive_feature_class_full_path, template_feature_class_full_path,
613                           data_zone_feature_class_full_path, country_feature_class_full_path, operating_system,
614                           data_source, database_host, database_port, database_name, database_user, database_password,
615                           horizontal_distance_threshold):...
1505
1506
1507 # Process Result Function
1508 # -----
1509 def process_result(result):...
1511
1512
1513 # Main body of the script
1514 # -----
1515 # Defines the entry point into the script
1516 def main():
1517
1518     # Capture script parameters
1519     # -----
1520     root_folder = sys.argv[1] # e.g., C:/att/project/apcshare/public/raster-ingest/ | /att/apcshare/raster-ingest/
1521     server_environment = sys.argv[2] # e.g., alpha | dev-nccs | prod-nccs | dev-drg | prod-drg
1522     linux_path = sys.argv[3] # e.g., /css/nga/
1523     windows_path = sys.argv[4] # e.g., P:/data/
1524     master_feature_class = sys.argv[5] # e.g., nga_footprint_master_v2
1525     archive_feature_class = sys.argv[6] # e.g., nga_footprint_archive_v2
  
```

The screenshot shows the Apache Guacamole web interface in a browser window. The address bar displays `guac.nccs.nasa.gov/guac/#/`. The user is logged in as `jshute`. The interface is divided into two main sections: **RECENT CONNECTIONS** and **ALL CONNECTIONS**.

RECENT CONNECTIONS displays six thumbnails of active sessions:

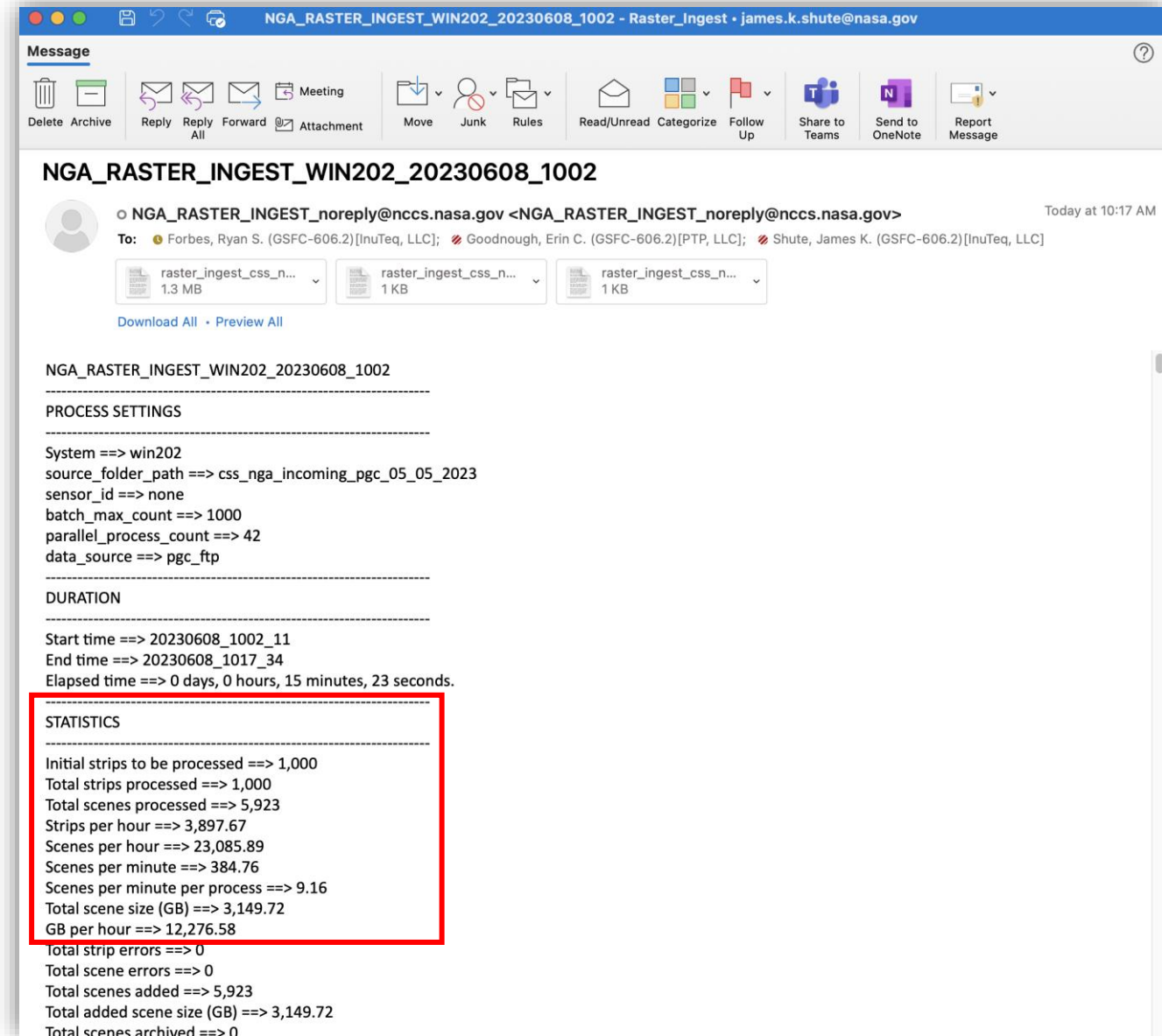
- `win202`: A terminal window with a file explorer overlay.
- `winadm202`: A blue desktop background with a terminal window.
- `win201`: A blue desktop background with a user login prompt.
- `win207`: A blue desktop background with a terminal window.
- `win210`: A blue desktop background with a user login prompt.
- `win206`: A terminal window with a file explorer overlay.

ALL CONNECTIONS displays a list of available sessions:

- win201
- win202
- win203
- win204
- win205
- win206
- win207
- win208
- win209
- win210
- winadm201
- winadm202

A status message on the right indicates: *Currently in use by 2 users.*

- Ingest rates are now 25k rows/hour/VM (as many as 10 VMs at the same time)
- Other processes utilizing this approach
 - Deleting rows or files
 - Duplicate record removal
 - Exporting rows from development to production (500-600k rows/hour)
 - Master Footprint (MFP) evaluation
 - Raster validation (100-200k rows/hour)



Message

NGA_RASTER_INGEST_WIN202_20230608_1002 - Raster_Ingest • james.k.shute@nasa.gov

NGA_RASTER_INGEST_WIN202_20230608_1002

o NGA_RASTER_INGEST_noreply@nccs.nasa.gov <NGA_RASTER_INGEST_noreply@nccs.nasa.gov> Today at 10:17 AM

To: Forbes, Ryan S. (GSFC-606.2)[InuTeq, LLC]; Goodnough, Erin C. (GSFC-606.2)[PTP, LLC]; Shute, James K. (GSFC-606.2)[InuTeq, LLC]

raster_ingest_css_n... 1.3 MB
raster_ingest_css_n... 1 KB
raster_ingest_css_n... 1 KB

Download All • Preview All

NGA_RASTER_INGEST_WIN202_20230608_1002

PROCESS SETTINGS

System ==> win202
source_folder_path ==> css_nga_incoming_pgc_05_05_2023
sensor_id ==> none
batch_max_count ==> 1000
parallel_process_count ==> 42
data_source ==> pgc_ftp

DURATION

Start time ==> 20230608_1002_11
End time ==> 20230608_1017_34
Elapsed time ==> 0 days, 0 hours, 15 minutes, 23 seconds.

STATISTICS

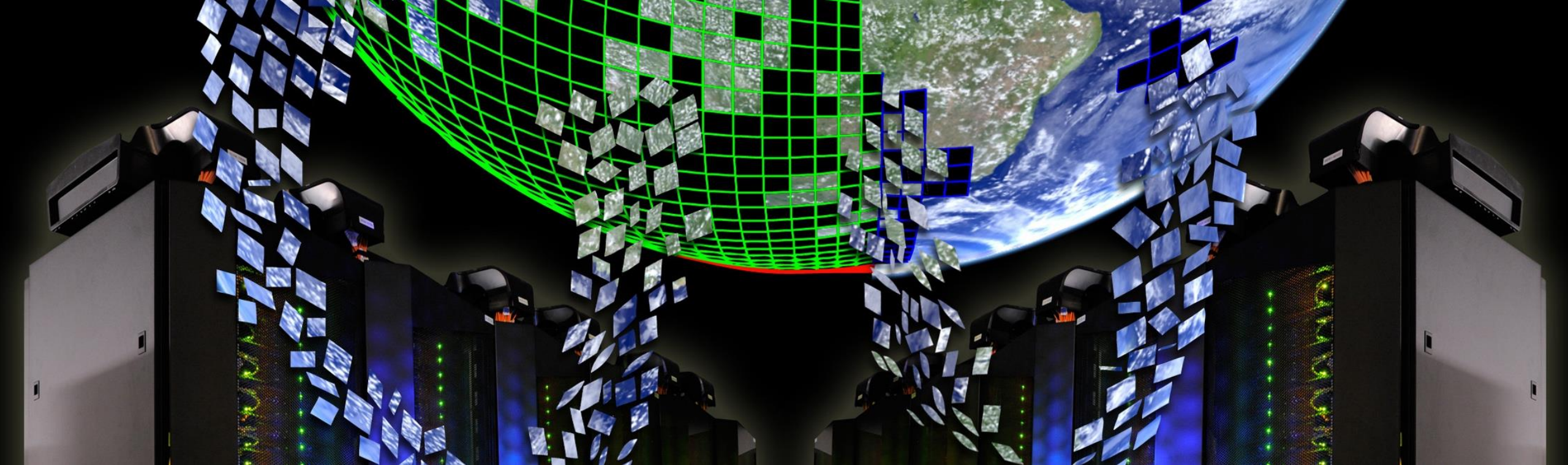
Initial strips to be processed ==> 1,000
Total strips processed ==> 1,000
Total scenes processed ==> 5,923
Strips per hour ==> 3,897.67
Scenes per hour ==> 23,085.89
Scenes per minute ==> 384.76
Scenes per minute per process ==> 9.16
Total scene size (GB) ==> 3,149.72
GB per hour ==> 12,276.58
Total strip errors ==> 0
Total scene errors ==> 0
Total scenes added ==> 5,923
Total added scene size (GB) ==> 3,149.72
Total scenes archived ==> 0

- Using an IDE (e.g., PyCharm) was instrumental in helping with the debugging process
- Build repeatable tasks into their own functions and leverage across the environment (e.g., email_notification function)
- Python multiprocessing pool.apply_async was the easiest approach to get working
 - Requires a callback function
- Use script parameters to segment the workload (e.g., sensor_id)
- Write the script to work in serial or parallel mode
 - Debugging does not work inside parallel functions/processes
- Issues may arise when attempting to run too many processes in parallel (e.g., bad/incorrect results or counts)
- Build in logging, statistics calculations, and email notifications
- Functions called from external/other files, global counts, and lists cannot be accessed by a parallel process
- A complex object (e.g., database connection) cannot be sent to a parallel function.

```
if process_method == "serial":
    # Use this function to test one record at a time
    # -----
    strip_result = process_raster_strip(strip_full_path, linux_path, windows_path,
                                      master_feature_class, master_feature_class_full_path,
                                      archive_feature_class, archive_feature_class_full_path,
                                      template_feature_class_full_path, data_zone_feature_class_full_path,
                                      country_feature_class_full_path, operating_system, data_source,
                                      database_host, database_port, database_name, database_user,
                                      database_password, horizontal_distance_threshold)

    # Append the strip_result from the individual strip to the result list
    # -----
    if strip_result:
        logger.info("Append the strip_result from the individual strip to the result list")
        result_list.append(strip_result)

else:
    # Use this function when parallel processing
    # -----
    pool.apply_async(process_raster_strip, args=(strip_full_path, linux_path, windows_path,
                                              master_feature_class, master_feature_class_full_path,
                                              archive_feature_class, archive_feature_class_full_path,
                                              template_feature_class_full_path,
                                              data_zone_feature_class_full_path,
                                              country_feature_class_full_path, operating_system,
                                              data_source, database_host, database_port, database_name,
                                              database_user, database_password,
                                              horizontal_distance_threshold), callback=process_result)
```

Questions?

Contact Information

NCCS Cloud Computing and Data Sciences (CCDS) Lead	Jim Shute (james.k.shute@nasa.gov)
NCCS GIS Manager	Ryan Forbes (ryan.s.forbes@nasa.gov)
NGA Data Manager	Erin Goodnough (erin.c.goodnough@nasa.gov)
NCCS Website	https://www.nccs.nasa.gov
NCCS Spatial Analytics Platform	https://maps.nccs.nasa.gov