

From Data to Models and Analytics

ESS CI Working Group

29-Apr-2019

John Tourtellott

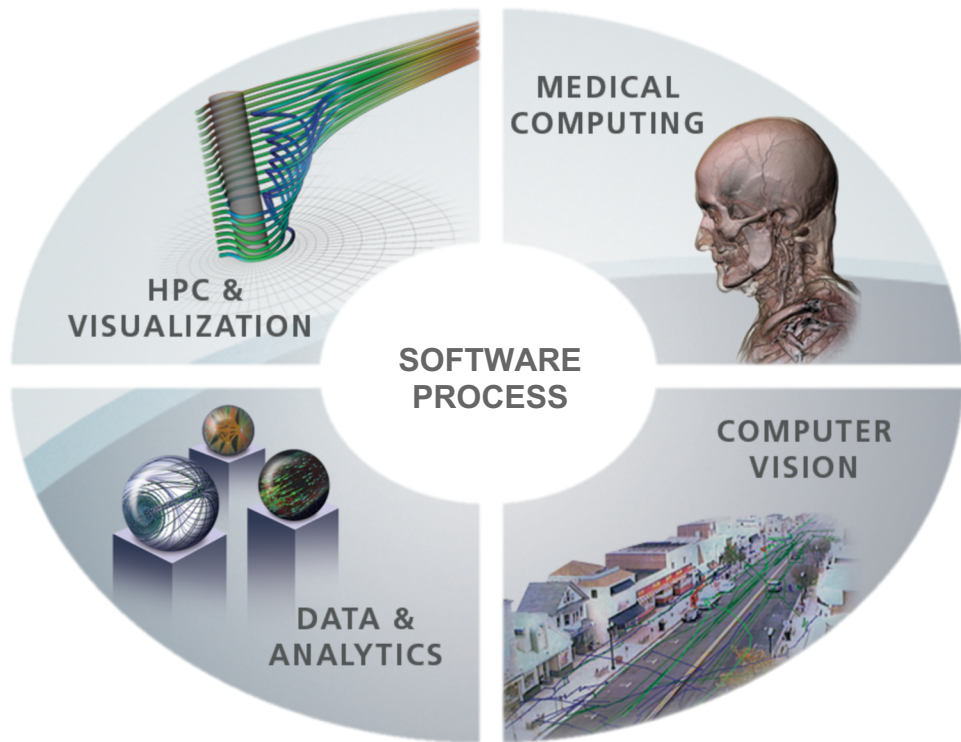


Collaborative software R&D

Technical computing
Algorithms & applications
Software process & infrastructure
Support & training
Open source leadership

Supporting all sectors

Industry, government & academia

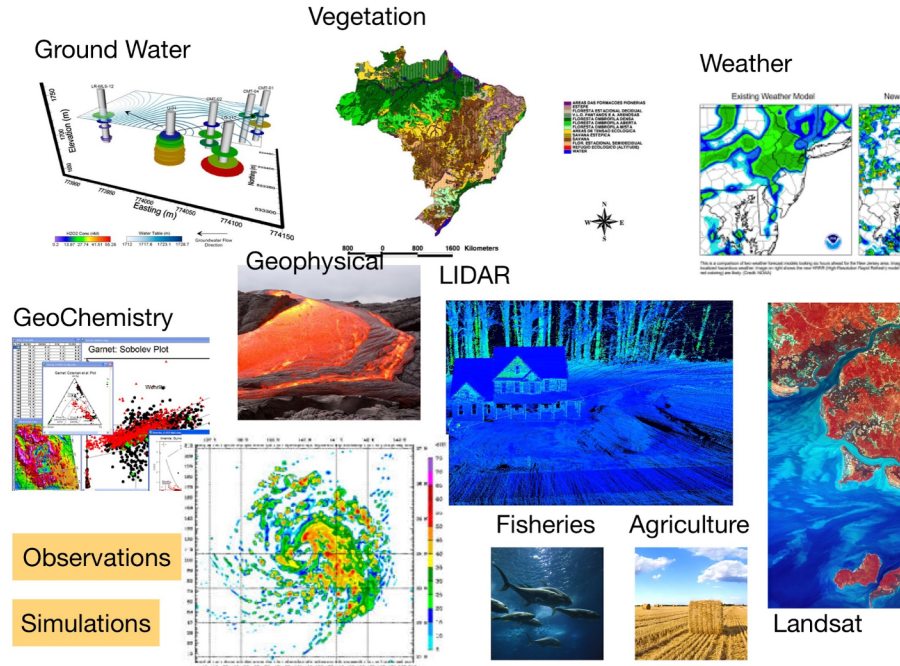


From Data to Models and Analytics

Incredible variety of data sources

- Spatially & temporally heterogeneous
- Varying data types and formats
- Nearly unlimited scale

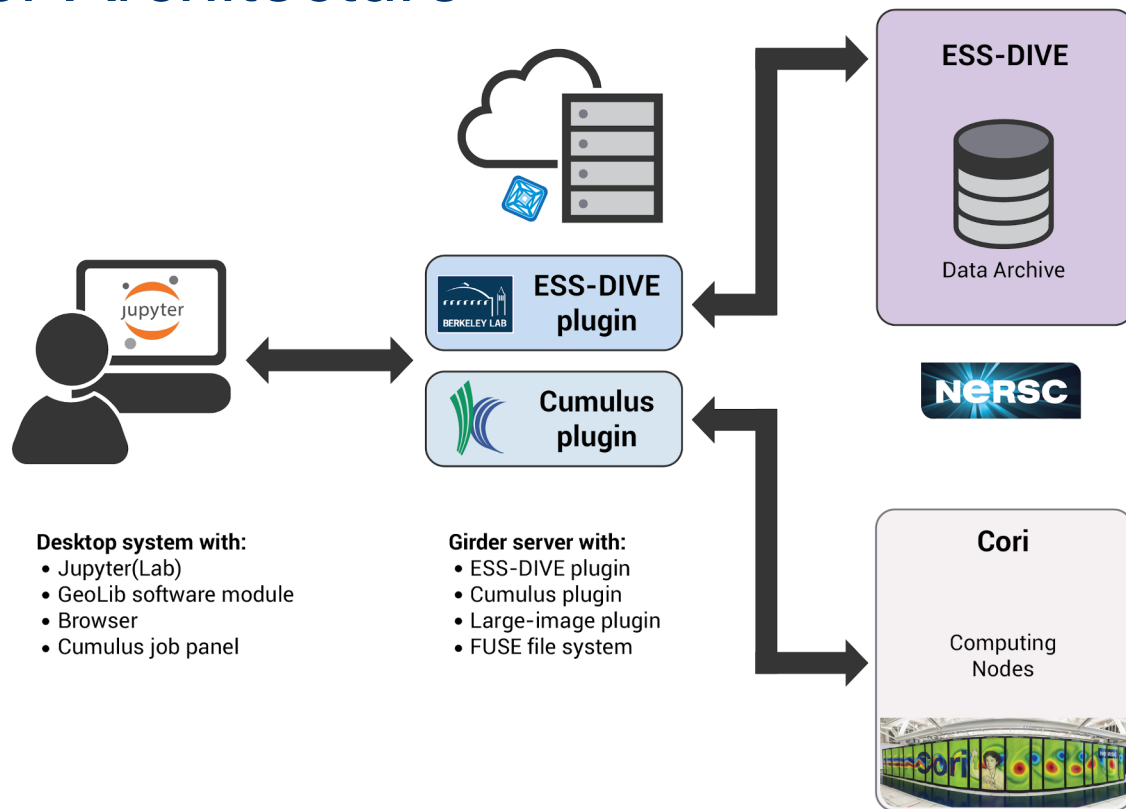
New modeling & simulation opportunities
Data preparation becoming a major effort



Simulation Modeling ToolKit (SMTK)

1. Unified data access
2. Minimal programming effort
3. Minimal data movement/footprint

Three-Tier Architecture



Minimal programming layer

```
geolib.create()
```

```
geolib.show()
```

```
geolib.crop(), geolib.reproject(), ...
```

```
geolib.save()
```

```
geolib.connect()
```

```
geolib.submit_crop(), ...
```

Jupyter Notebook

1. Authenticate to NERSC
2. Select dataset from ESS-DIVE
3. Define crop geometry
4. Submit processing job to NERSC (Cori)
5. (when complete) Display results
6. (optional) Save to local disk

File Edit View Run Kernel Tabs Settings Help

docs > examples

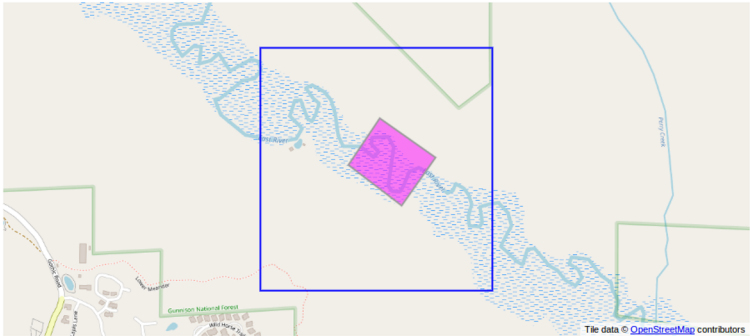
Name	Last Modified
ess_dive.ipynb	a month ago
gaia_processes.ipynb	5 months ago
gaia_show.ipynb	4 months ago
girder_large_image.ipynb	a month ago
girder_raster_crop.ipynb	a month ago
job_submit.ipynb	20 minutes ago
lidar_vectors.ipynb	4 months ago
nersc_login.ipynb	23 minutes ago
pygeojs_example.ipynb	4 hours ago
crop.py	4 months ago
gaia_processes.html	9 months ago
girder_example.py	5 months ago
job_submit.slides.html	3 minutes ago
watershed-crop.geojson	15 days ago

3. Define Crop Geometry

```
[6]: # Create polygon to display bounds
bounds = essdrive.object.get_metadata().get('bounds')
coordinates = bounds.get('coordinates')
import geojson
bounds_geometry = geojson.Polygon(coordinates)
bounds_feature = geojson.Feature(geometry=bounds_geometry, properties={'fillOpacity': 0, 'strokeColor': 'blue'})
bounds_object = geolib.create(bounds_feature)

# Get crop geometry
crop_object = geolib.create('watershed-crop.geojson')

scene1 = geolib.show([crop_object, bounds_object])
display(scene1)
```



... 15.50

4. Submit preprocessing job on NERSC (Cori)

pygeojs_example.ipynb	4 hours ago
crop.py	4 months ago
gala_processes.html	9 months ago
girder_example.py	5 months ago
watershed-crop.geojson	15 days ago

4. Submit preprocessing job on NERSC (Cori)

```
[8]: # To run on local machine or Girder:
# output_object = geolib.crop(essdive_object, crop_object)

# To run on NERSC:
import getpass
while not nersc_repository:
    nersc_repository = getpass.getpass('Enter NERSC repository (account): ')
cori_job = geolib.submit_crop(essdive_object, crop_object, nersc_repository, job_name='watershed')
cori_job

Enter NERSC repository (account): .....
user {'login': 'johnt', 'groupInvites': [], 'modelType': 'user', 'emailVerified': True, '_accessLevel': 2, 'firstName': 'John', 'public': True, '_id': '5a60e9de0640fd01195132e4', 'status': 'enabled', 'admin': True, 'groups': [], 'email': 'john.tourtellott@kitware.com', 'otp': False, 'size': 3003957, 'created': '2018-01-18T18:39:26.139000+00:00', 'lastName': 'Tourtellott'}
Creating cluster on cori
Creating SLURM script watershed
script_id 5cabel66ed1b635fb960c795
Creating job watershed
Created job folder e09f51ec3b9f4ed190d347f866ce1a52
Created job_id 5cabel66ed1b635fb960c799
Uploading geometry file
Submitting job
submit_job body: {'queue': 'debug', 'account': 'm2690', 'jobOutputDir': '/global/cscratch1/sd/johnt/geolib/190408/watershed', 'maxWallTime': {'hours': 0, 'seconds': 0, 'minutes': 5}, 'numberOfNodes': 1, 'constraint': 'knl', 'machine': 'cori'}
Submitted job 5cabel66ed1b635fb960c799

[8]: '5cabel66ed1b635fb960c799'
```

5. Use job id to create GeoLib object

cumulus

File Help

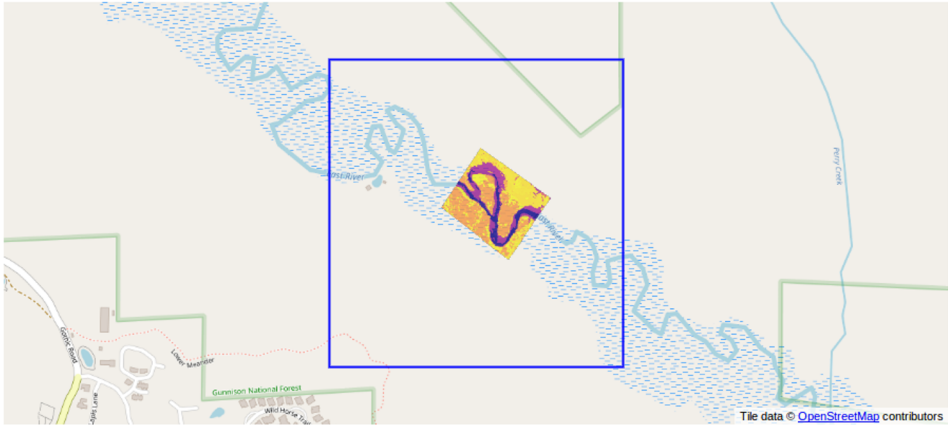
Job Id	Machine	Job Name	Nodes	Cores	Status	Started	Finished	Notes
5c9540c8ed1b6319b488e4ad	cori	geolib	1	1	complete	22-Mar-2019, 16:08	22-Mar-2019, 16:15	
5c953c3fed1b6319b488e478	cori	geolib	1	1	complete	22-Mar-2019, 15:49	22-Mar-2019, 15:54	
5c953c3aed1b6319b488e46d	cori	geolib	1	1	complete	22-Mar-2019, 15:49	22-Mar-2019, 15:54	
5c9533feed1b6319b488e445	cori	geolib	1	1	complete	22-Mar-2019, 15:14	22-Mar-2019, 15:19	
5c94ec82ed1b6319b488e415	cori	geolib	1	1	complete	22-Mar-2019, 10:09	22-Mar-2019, 10:13	
5c94012ced1b6319b488e3fd	cori	geolib	1	1	complete	21-Mar-2019, 17:25		
5c93c50ded1b6319b488e3dc	cori	geolib	1	1	complete	21-Mar-2019, 13:08		
5c939fefed1b6319b488e397	cori	geolib	1	1	complete	21-Mar-2019, 10:30		
5c92a5c4ed1b630121d67bb4	cori	geolib	1	1	complete	20-Mar-2019, 16:42		
5c92880ced1b630121d67b94	cori	geolib	1	1	complete	20-Mar-2019, 14:35		

File Edit View Run Kernel Tabs Settings Help

neresc_login.ipynb job_submit.ipynb Python 3

5. Use job id to create GeoLib object

```
...  
[58]: job_id = 'Scab97f3ed1b635fb960c735' # (PH_community_distribution_map.tif)  
  
girder_url2 = girder.lookup_url(job_id=job_id)  
output_object = geolib.create(girder_url2, bounds=crop_object.get_data().bounds.values[0])  
# print('output_object', output_object)  
  
# Display cropped dataset  
output_object._epsg = 4326  
output_object._setdatatype(geolib.types.RASTER)  
output_object.opacity = 0.8  
output_object.set_mapnik_style({  
    'band': 1,  
    'max': 65,  
    'min': 10,  
    'palette': 'matplotlib.Plasma_6',  
    'scheme': 'linear',  
})  
  
scene2 = geolib.show([output_object, bounds_object])  
display(scene2)
```



Tile data © [OpenStreetMap](#) contributors

Simulation Modeling ToolKit (SMTK)

1. Unified data access
2. Minimal programming effort
3. Minimal data movement/footprint

Open Source, Open Science

<https://github.com/OpenDataAnalytics/gaia>

<https://github.com/OpenGeoscience/geojs>

<https://github.com/OpenGeoscience/pygeojs>

<https://github.com/girder/girder>

https://github.com/girder/large_image

<https://github.com/Kitware/cumulus>

<https://github.com/shreddd/girder-ess-dive>