CS 205 Project - M4

Parallelizing Crop Phenology reports via NDVI datasets

Team 18:

Elaine Swanson, Alex Johnson, Saketh Mynampati, Alyssa Ross

Sequential Baseline:

```
+--------------------+
|     Functions      |
+--------------------+

exec(command)
downloadFile(query, filename)
writeDataToFile(data, filename)
outputProfile(counters, dt)


+--------------------+
|       main()       |
+--------------------+

  Initialize maps
```

```
+----------------------+
|   Loop over years    |
|  (2001 to USER DEFINED) |
+----------------------+
```

Check for leap years
Loop over months and days
Calculate week number
Iterate over region codes
Construct download URL and filename
Download file
Initialize Geotiff object
Define functions for reading data
Initialize profiling variables
Profile reading files
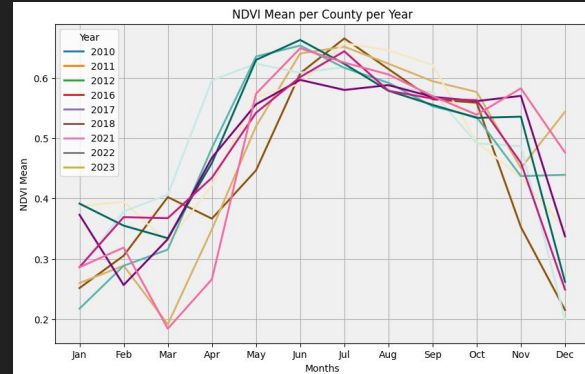Process data and calculate <u>statistics</u>
Perform <u>SSA</u>
Write output to files
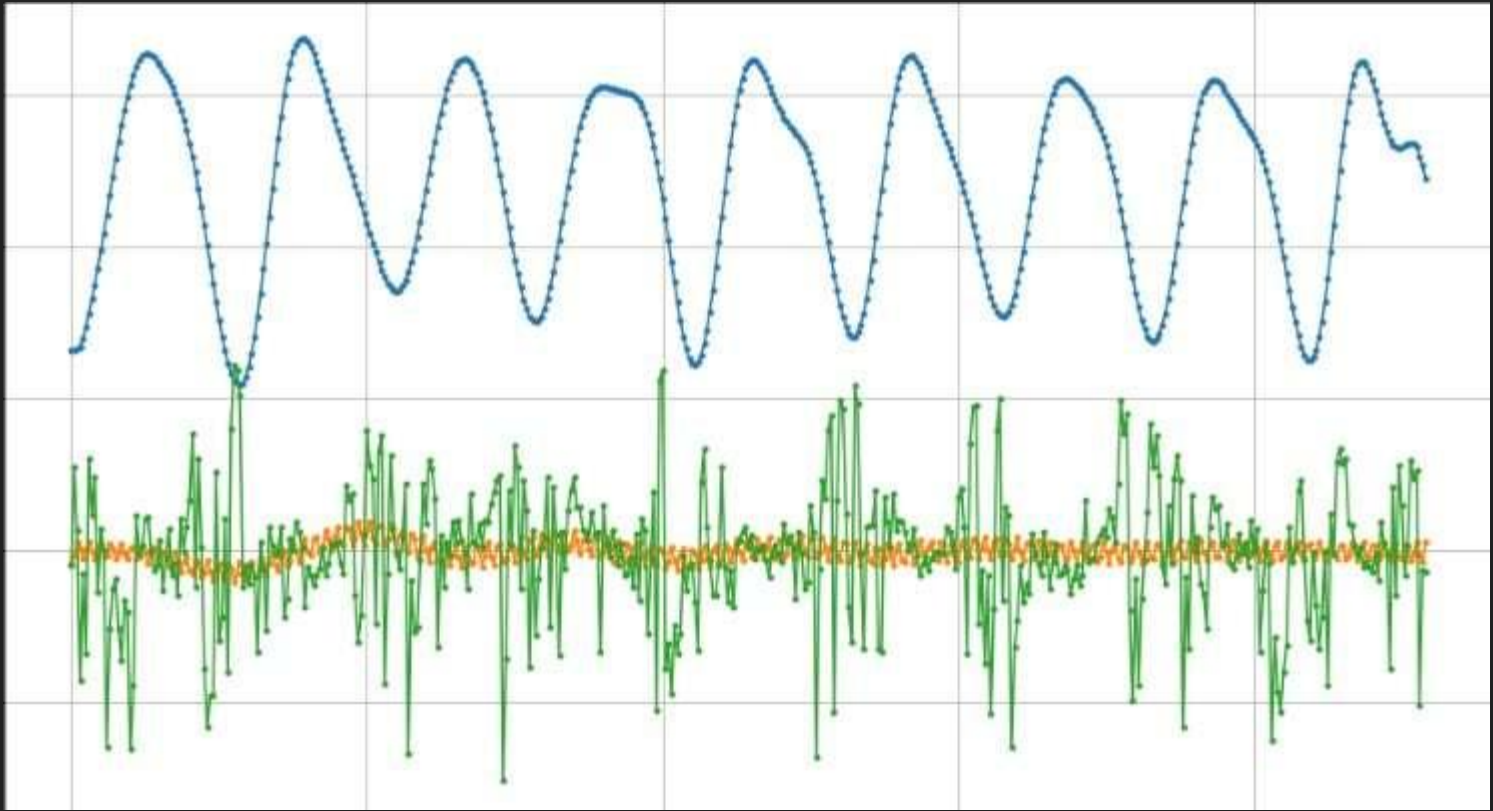Output profiling information
End program

NDVI Image
20120618 to 20120624

NDVI Mean per County per Year
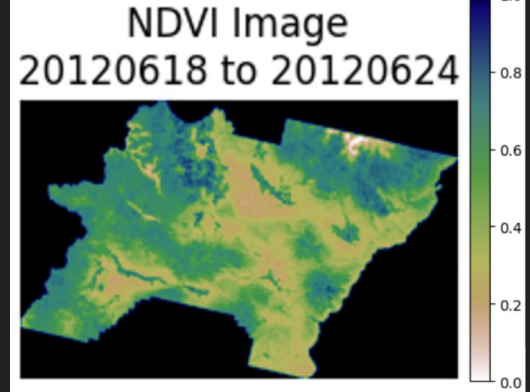
<u>Singular Spectrum Analysis
(SSA) Matrix Binary File
Example: (257, 409)</u>

Sequential Baseline:

Execution time of different sections
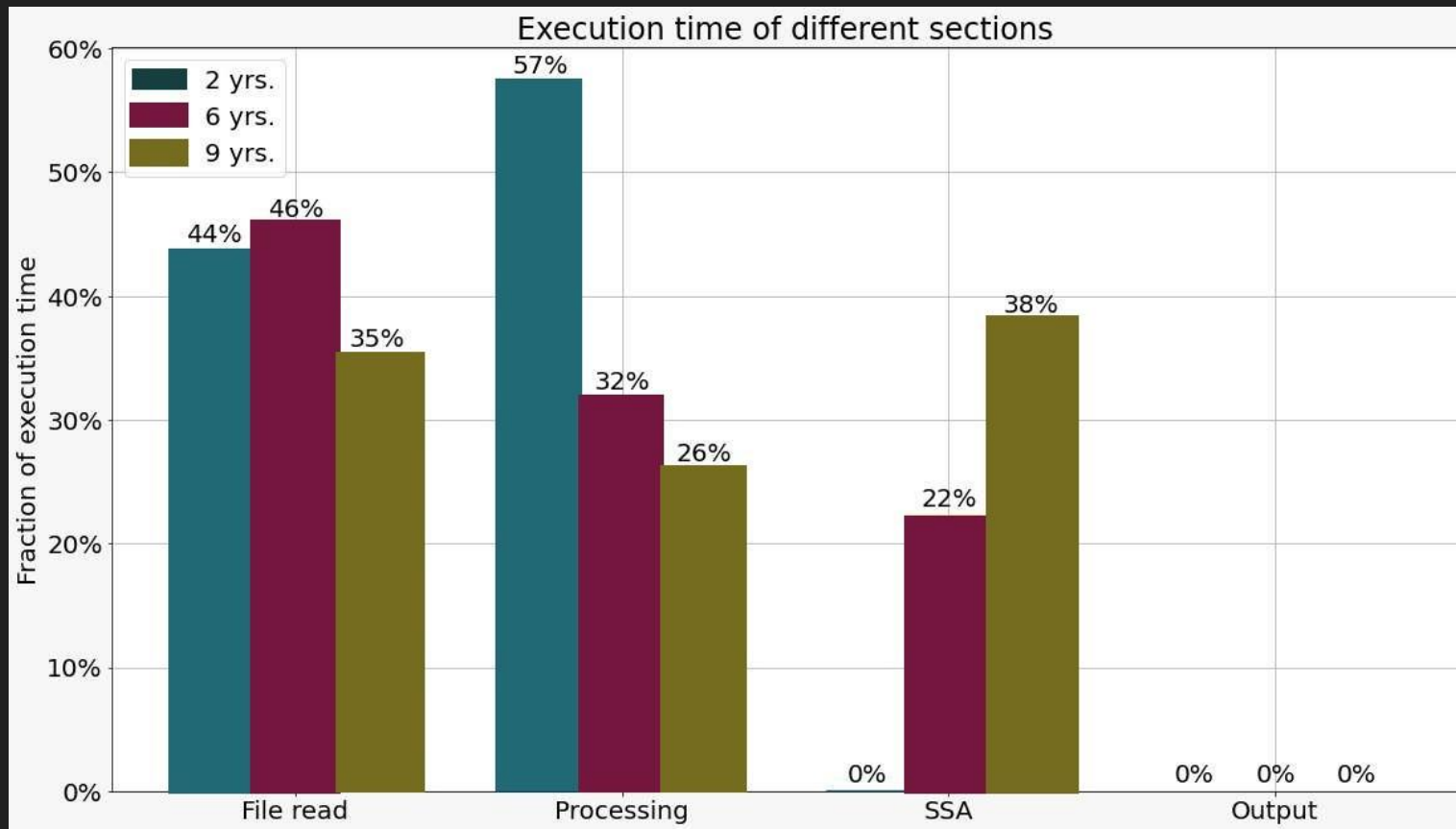
===== **Processor Information** =====

Physical/Logical Cores: 18/36

Frequency (GHz): 3.40

===== **Memory Information** =====

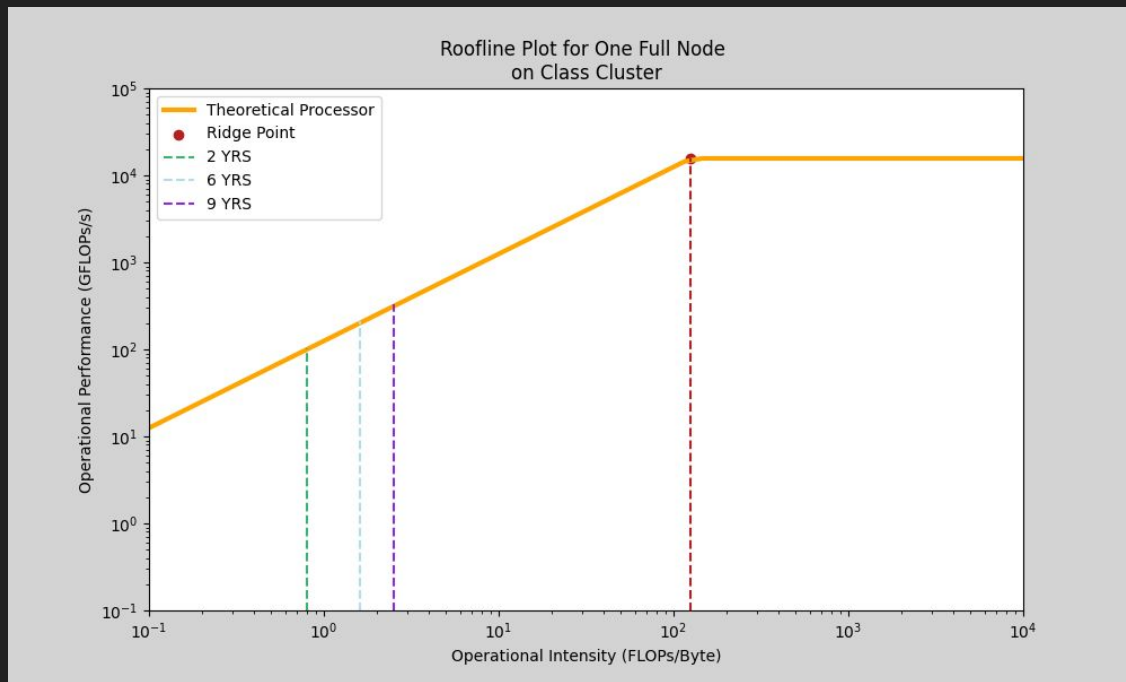**Total Memory (GB): 68.57**

Available Memory (GB): 43.61

===== **Disk Information** =====

Device: NVMe storage device

> **Total Space (GB): 49.99**
>
> **Free Space (GB): 29.68**

Roofline Plot for One Full Node
on Class Cluster

Peak Floating Point Performance (Integer Precision): 15674 GFLOPs

Peak Memory Bandwidth: 125 GB/s

The ridge point I_β is: 125.392

Proposed Parallelization:

- **Data: x, y (county) and week/year.**
- **Threads:**
  - **Shared disk, shared I/O limitations.**
- **Main Thread: Coordinates all other OMP threads (36).**
  - **GOAL: Cropland data is separate from NDVI data, so threads handle loading and overlaying of NDVI and Crop Masks**
  - **Three kinds of parallelization to test:**
  - **Simple**
    i. **Each thread downloads, overlays, and processes data for a given county, with 1 or 2 threads per county.**
    ii. **Two-thread counties communicate via MPI non-blocking operations to minimize waiting - each handles a portion of time.**
    iii. **Synchronize at end within county, MPI File Ops for output.**
  - **Maximize Cache Hit Rate/Resource Use:**
    i. **All threads handle single county at a time, and apply mask in sequence.**
    ii. **File is resynthesized on single assigned thread via MPI_Reduce for summary statistics over time.**
    iii. **Built into time series over chosen years using MPI**
    iv. **Heavy MPI communication overhead.**
- **Note:**
  - **Program still lies in the memory-bound region of the roofline model.**